Pattern Signature Classifier SP4

User Manual

Contact Details

R.T.E.M. Limited
Unit 3
Leaches Farm Business Centre
Bicester Road
Kingswoood
Aylesbury
Bucks
United Kingdom

HP18 0RF

Telephone: +44 (0)1296 770765
Facsimile: +44 (0)1865 451477

Web : www.rtem.co.uk
Email sales@rtem.co.uk

SP4 Installation and User Manual

R.T.E.M © Copyright 2005-2008

(This document may not be reproduced without the strict permission of R.T.E.M. Ltd)

Document Control

Document Number	**:**	**UM00001**
Issue Number	**:**	**1.06**
Authors	**:**	**Anup Lekshmanan, Brian Moss**

Document History

Issue	Date	Description
1.0	20/02/2007	Draft Issue
1.01	14/10/2007	Draft issue 01
1.02	15/10/2007	Draft issue 02
1.03	30/10/2007	Draft Issue 03
1.04	10/03/2008	Draft Issue 04
1.05	26/06/2008	Draft Issue 05
1.06	31/08/2008	Draft Issue 06

CONTENTS

CHAPTER 1 INTRODUCTION 7
1.1 Counter Classifiers 7
1.2 Scope 7
1.3 Installation Training 7
1.4 Safety 7

CHAPTER 2 OVERVIEW 10
2.1 The Classifiers 10
2.2 Data Collection 10
2.3 Hardware and Software Configuration 10
2.4 Classifier Front and Rear view 13

CHAPTER 3 CLASSIFIER FUNCTIONS 18
3.1 Loop Signature Detection 18
3.2 Vehicle Processing 18
3.3 Classification 21
3.4 Data Storage 22
3.5 Data Transmission 24
3.6 Faults 24
3.7 Communications 24
3.8 Local Terminal 25

CHAPTER 4 COMMUNICATION PROTOCOLS 26
4.1 ASCII Terminal Commands 26
4.2 ASCII Terminal Command Description 29

CHAPTER 5 BINARY PACKET/MESSAGE PROTOCOL 62
5.1 Scope 62
5.2 Data link layer (RACCOM packet protocol) 62
5.3 MESSAGE Layer 63

CHAPTER 6 MULTIDROP NETWORK 72
6.1 Address links & Lane mapping 72
6.2 Operational Settings 73
6.3 Time & Date Broadcast synchronisation 73
6.4 Data storage 73
6.5 Polling rate vs. Vehicle rate 73

CHAPTER 7 THE MICRODIAL SOFTWARE 74
7.1 Overview 74
CHAPTER 8 INSTALLATION 75
8.1 Classifier Installation 75
8.2 Procedure 75
CHAPTER 9 COMMISSIONING AND TESTING 76
9.1 Essential Parameter check 76
9.2 Site Operational check 77
CHAPTER 10 MAINTENANCE 78
10.1 Storage 78
10.2 Operational 78
CHAPTER 11 TROUBLESHOOTING 79
11.1 Scope 79
11.2 Communications and Connections 79
11.3 Loops 79
CHAPTER 12 TECHNICAL SPECIFICATION 80

Chapter 1 INTRODUCTION

1.1 Counter Classifiers

This document provides operational information for the SP4 classifier and supported firmware solutions. Unless where explicitly specified the instructions are common for all firmware solutions.

The SP4 fitted with AXD4 firmware functions as an axle detecting counter classifier. It operates on two loops per each lane.

The SLC4 is a firmware variance, which operates on a single loop for each lane.

1.2 Scope

This document is the Installation and User Manual for the SP4 classifier and firmware variants designed and manufactured by R.T.E.M Limited.

This manual provides information required for installation, initial configuration and first level fault-finding.

⚠ WARNING

WARNING *notices alert you to risks of personal injury.*

⚠ CAUTION

CAUTION *notices alert you to risks of equipment damage or loss of data.*

1.3 Installation Training

Installation Engineers should be suitably qualified, or have previous experience in Roadside equipment installation.

⚠ WARNING

Be aware of potential hazards normally associated with any road side installation.

As a minimum, engineers must be familiar with:

1. *Safety aspects concerning equipment connected to telecommunications lines*
2. *Highway safety procedures*

1.4 Safety

This section identifies hazards that may be associated with installing this equipment.

The equipment itself is low voltage and low weight and does not pose any direct hazard. The installation of this equipment at roadside does warrant awareness of roadside safety procedures and should be only be carried out by properly trained staff.

1.4.1 Power DC Connection

This equipment uses DC power in the range 4 volts DC to 18 Volts DC. Operating the equipment outside this range can cause equipment failure, damage to the equipment or loss of data.

⚠ WARNING

DO NOT CONNECT MAINS

1.4.2 Loop Connection

Equipment can be connected to 4 loop inputs. Unused loop inputs should be left open circuit.

⚠ CAUTION

- *Do not short circuit loop inputs.*

1.4.3 Darlington Pair Output

Equipment can be connected to an external device, such as vehicle sign or camera.

⚠ WARNING

DO NOT CONNECT TO MAINS DEVICE

1.4.4 Highway Network Lines

This equipment can be connected directly to the longitudinal communications cables carrying communication signals conforming to RS485 which may be found on Highways. In common with other long distance telecommunications lines, these cables may present a risk of induced voltages. When installing networked systems on highways the appropriate care should exercise at all times. (Refer EN 60950-1: 2001 Clause 6).

1.4.5 Static Electricity

Electrostatic discharge (ESD) may present a risk of damage to delicate electronic devices. Always take suitable precautions to avoid ESD damage to equipment, particularly through improper handling methods.

⚠ CAUTION

- *This unit does not contain any user serviceable parts. Standard ESD precautions should be observed when handling the unit with the cover removed..*

Chapter 2 OVERVIEW

2.1 The Classifiers

The SP4 vehicle classifier is an advanced electronic device that is used in roadside cabinets connecting to inductive loops. Two loops are necessary for a speed/classification site together with the AXD4 firmware. Each SP4/AXD4 can be connected to a maximum of 4 loops to give operation on 2 lanes. The SP4/AXD4's can be connected together to monitor and record up to 14 lanes (optionally 16 lanes).

If the SLC4 firmware is installed the SP4 will give similar results with the exception that it uses a single loop to monitor and record vehicles. This allows the unit to monitor up to 4 lanes. The units can be connected together to monitor up to 28 lanes.

The SP4/AXD4 can operate on a number of classification schemes, whilst the AP4/SLC4 currently only operates on EURO-6.

2.2 Data Collection

The primary function of the Classifier is to record individual vehicle records. Each vehicle is classified according to length, number of axles, axle separation and relative chassis height from the ground. In addition the classifiers pattern matches the signature. Pattern matching provides a unique way to define a signature thus increasing the classification accuracy.

The single loop classifier measures the speed and length from signature derived from the single loop. The information derived is passed through a complex statistical analysis engine to derive the speed and length.

The SP4/SLC4 uses the traditional two vehicle loop event triggers to calculate the vehicle speed and length.

The classifier can additionally store vehicle signatures as well as processing vehicle individual records.

2.3 Hardware and Software Configuration

All the functions listed above are implemented in hardware and software within the Classifier.

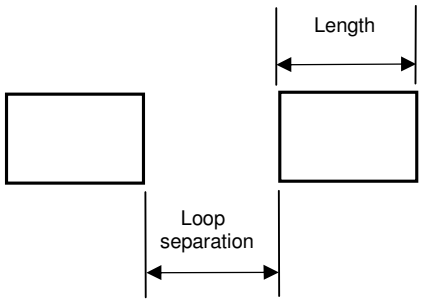
2.3.1 Hardware

2.3.1.1 Loop Connection

Each unit allows up to 4 loops to be connected. The loops dimension and physical layout should be within the flowing range

Loop length 10cm < range < 600cm

Loop separation 10cm < range < 600cm



With SP4/SL loops must be connected in pairs, whilst the SP4/SLC4 has loops connected individually.

Loop Pairs

Lane 1	Loop 1 and Loop 2
Lane 2	Loop 3 and Loop 4

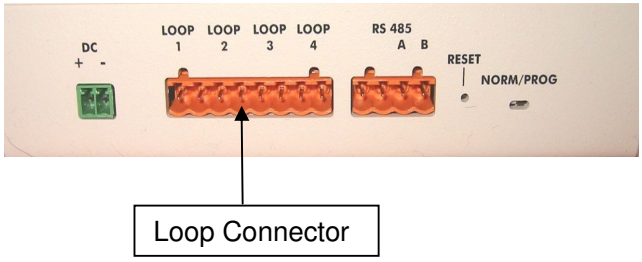
For more than 2 lanes refer to the table in section 6.1 for the internal address link connection and loop mapping.

For straddle algorithm loops are paired for each adjacent lane. These are referred to as straddle pairs.

Lane 1	Lane 2
Loop 1	Loop 3
Loop 2	Loop 4

Straddle loop pairs are loop 1 and loop 3, and loop 2 and loop 4. These must be adjacent to each lane. Straddling vehicles are those that run over both loops, example driving in the centre of the road.

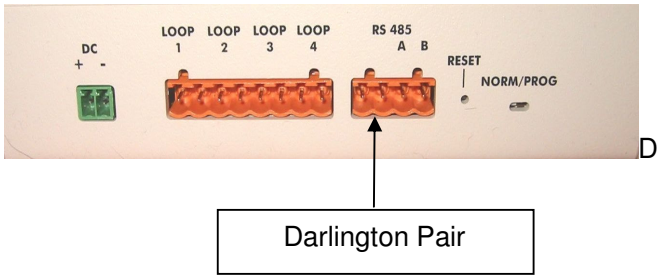
Straddle pairing should be observed when installing the unit on the road.



2.3.1.2 Darlington Pair Output

The unit provide a Darlington pair output that can sink up to 1A steady state at a maximum voltage of 12 Volts DC. The ground must be common between the unit's power and the source.

In the AXD4 firmware this output is used to control the supply to an external modem to reduce power consumption. The output may also be used for other switching applications which may include functions such as switch a sign or camera on, in red-light running scenarios or bus lane enforcement.

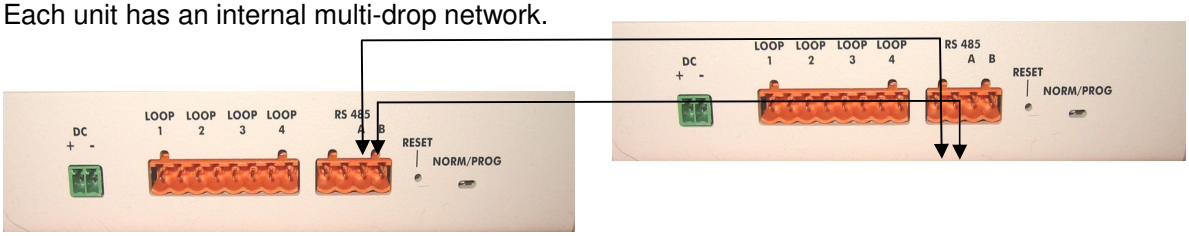


D

Darlington Pair

2.3.1.3 Network

Each unit has an internal multi-drop network.



RS485 Network Pair, Connect A-A and B-B up to 7 devices

When connecting as a multi-drop, each unit must be assigned a unique address – refer table in section 6.1

2.3.2 Software

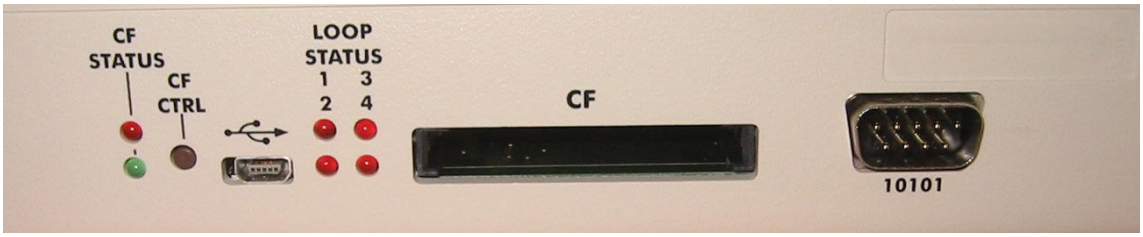
The unit firmware is downloaded from a PC through a standard USB port. The download process will copy the firmware to the non-volatile internal flash memory. (Note this

is not the compact flash, which is external non-volatile memory). The internal flash size is 390K.

The internal flash can be programmed between 100 and 1000 times. The download procedure is detailed below:

2.4 Classifier Front and Rear view

2.4.1 Front view



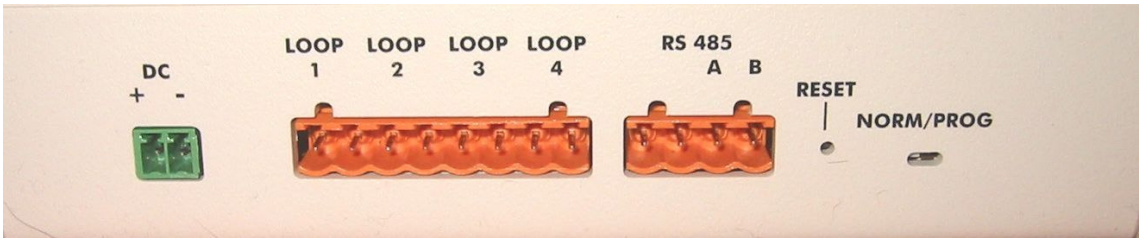
CF LED	Red LED
	When the **CF CTRL** switch is pressed and if the CF card is saving data into a file, the Red LED will turn on for up to a second.
	Every time the data is written to the CF card the LED will flash
Status	Green LED.
Green LED	When Powered on it will flash every 2 seconds. If the CF card is not plugged in on Power up it will stay on for initial 10 seconds before reverting to normal flash.
(Heart Beat)	
	After 10 minute the LED's will shutdown. Led indicators will turn on if the Push button is pressed.
CF	Pressing momentarily will enable the led indication for approximately 10 minutes.
CTRL	
Push button Switch	If the user presses and holds this button for approximately 5 seconds the loop status LED's will flash in a 'X' pattern and the user can safely replace the CF card.
USB Modem Connector	The USB-B mini port allows direct connection to any host PC USB port.
Loop Status	This normally Indicates loop presence which each loop coming on as a vehicle passes the loop
LED's	
	If all 4 led's are flashing then the module is in a default state.
	If the 'X' pattern is displayed then it is safe to remove the CF card and replace it.
	The LED's switch off after approximately 10 minutes to conserve power.

CF Card slot	Slot in Compact flash cards To hot swap, press and hold the approximately 5 seconds. The LED's will show a 'X' pattern. Remove and replace. Hard swap, after removing the card please wait 60 seconds before re-inserting the card.
RS232 Modem – 9way D type connector	This is a full handshake modem port. Connect to the modem using the standard cable supplied by the modem manufacturer.

2.4.2 Rear Panel of the Classifier

The figure below displays the classifier rear connections as detailed in the following table.

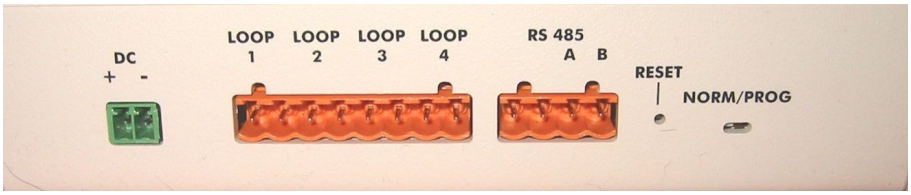
The Loop Connectors comply with Highways Agency MIDAS TR2169 Outstation specification standard.



DC Power input	This is a 2 pin connection for the DC input supply. This must be in the range 4 – 18 volts DC.
Loop Connector	One 8-way connector, which connects to the Loop Interface cable. ⚠ **CAUTION** It is ***essential*** to maintain the connection order of loop relative to the monitoring lane, otherwise the loop signals will not be interpreted incorrectly and vehicles will not be detected. Lanes are assigned : lane 1 : loop 1 & 2 Lane 2 : loop 3 & 4
RS485 Network	The 2 - Pins label A and B are used for the RS485 multi-drop

connector/ Power switch	connection. The two left hand pins are used for the optional power switch for external Modems.
Reset switch	This is only accessible using a small pin. The reset is a warm reset if the CF card is plugged in, otherwise it will result in a full coldstart.
Normal/Program Switch	This switch allows the firmware to be upgraded. To upgrade the internal firmware this switch must be placed in the 'Prog' position. **⚠ CAUTION** *It is essential to position this switch to 'Normal' operation position, otherwise the classifier will not work.*

2.4.3 How to download new firmware

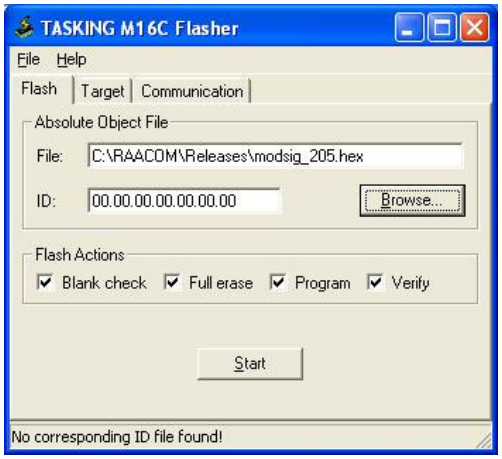


1 - Move the dip switch at the rear to right - PROG position

2 - Press the reset button (AT THE REAR using a pin)

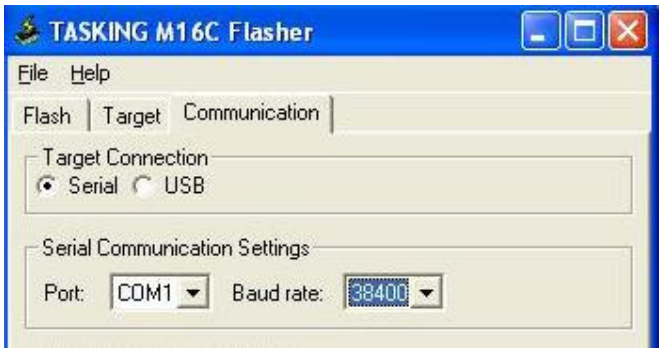
Run flasm16c.exe, which is third party software freely available from the internet.

1. Select the 'Flash' tab and tick all check boxes. Using the browse select the appropriate firmware .hex file (as shown below).



2 Select the 'Target' tab and ensure is M16C20-M16C60 as shown below.

3. Select the communication tab and set the target connection to serial. Select the appropriate com port and set the baud rate to 38400.



4. From the main Flash tab press the start button.

At the end of the download ensure that the dip switch is re-positioned to ‘NORMAL’ Press the reset button.

2.4.3.1 Firmware

Firmware is named as
SP4– MODSIG<VERSION>.HEX
Example: - MODSIG_205.HEX. The version is 2.05

SP4/SLC4 – SLP<VERSION>.HEX

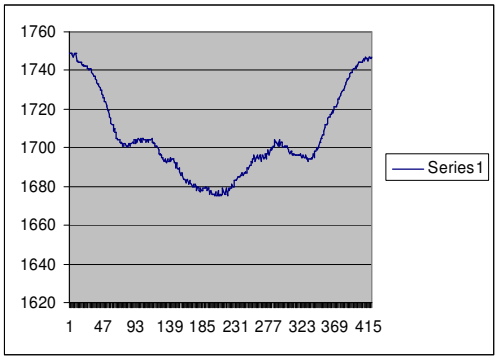
CYC400 – CYCLEDET<VERSION>.HEX

Any of the firmware can be downloaded to the same hardware using the above method.

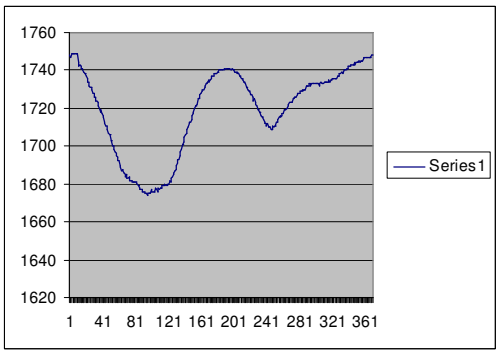
Chapter 3 CLASSIFIER FUNCTIONS

3.1 Loop Signature Detection

Every vehicle exhibits a unique magnetic signature, which can be picked up by small changes of inductance as the vehicle passes over it. These signatures are directly related to the metal shape of the vehicle and how these metal parts are positioned relative to each other in a 2 dimension space (i.e. height and length separation). Hence for each vehicle the signature can be clearly seen and identified. The graph below shows the signatures of a Bus and a Car+Trailer.



Signature of a bus.



Signature of a Car with a Trailer

The traditional loop detector measures the change of the frequency due to inductance change due to a vehicle passing over it, and compares this to a threshold value to determine a presence and the duration of this presence. With both versions of firmware the unit captures the waveform and analyses it real time in addition to determining the presence.

The signature analysis module also employs dual tracking of the reference unlike many traditional tracking. This allows for constant automatic tuning resulting in greater measurement accuracy. Hence unlike the traditional loop detectors where the percentage change from the datum value is required to trigger a vehicle event, the signature analysis reviews the signal itself to determine the presence of the vehicle. This is a powerful methodology which allows detection of very slow moving vehicles. It also means the algorithms employed are not as overtly dependant on quiescent inductance level. Furthermore, using the dual tuning mechanism the algorithm is always tracking. The end result is exceptional responses with queuing traffic and also from traditionally more difficult sites.

Each loop scanning can be adjusted by the unit. Crosstalk and inter-loop interference are avoided by fast multiplexing. For each loop 2Kbytes per second signature data are generated, which are processed in real time by the integrated high speed processor. This ensures the system does not suffer from having additional latency introduced by the loop detector being separate from the main CPU core as in other manufactures products

3.2 Vehicle Processing

Individual loop events are subjected to amplitude and differential analysis algorithms. The events provide details including the number of axles, axle separation, and 'air gap'

separation, and the type of signature. Each signature for cars, lorries, buses, HGV's etc have unique characteristics. These characteristics are summarized to provide a best match to typical signature vehicle class group. The loop events are processed to define and measure the vehicle type, speed length etc.

For the dual loop SP4 two loop events are required before vehicle processing is completed (with the exception of motor-bikes). SP4 dual loop vehicle processing uses traditional method to evaluate the speed and length. Speed and length is derived from the on/off events of the leading and trailing loop. If the events are not overlapping, the SP4 will check if the events belong to a motorbike. Non overlapping events that are not identified as matching a motor bike are not processed.

With the Single loop each single loop events are processed individually. Events from loops are used to identify the vehicle and calculate the speed and length.

The table below shows the VBV definition

VBV Field definition

Std	Field	Description
Y	**Date**	The date format YYYY-MM-DD
Y	**Time**	Time 24 Hour time format HH:MM:SS (00:00:00 to 23:59:59)
Y	**Vehicle record number**	A unique sequential number range 0-99 999 999. (100 million vehicles)
Y	**Lane number**	SP4/SLC4: Each lane contains a pair of loops, Lane 1 : is formed by loop 1 and loop 2 Lane 2 : is formed by loop 3 and loop 4 If an address header is put inside board, the lane = lane+ (offset*2), i.e. lane number is added to the address offset. Address offset '0' is always for the master device. Offsets 1 to 7 are for slaves. Hence slave 1 will identify lane 3 and 4 etc, Daisy chained SP4/SLC4's can have 16 lanes SP4/SLC4 : Each lane contains a single loop Lane 1 is loop 1 Lane 4 is loop 4 If an address header is put inside board, the lane = lane+ (offset*4), i.e. lane number is added to the address offset. Address offset '0' is always for the master device. Offsets 1 to 7 are for slaves. Hence slave 1 will identify lanes 5 to 8 etc, Daisy chained SLP can identify up to 32 lanes.
Y	**Length**	Range of length is 1 cm to 65,535 cm inclusive, in steps of 1 cm. The length

Std	Field	Description
		equations are given below. `length = (Dspeed *( LeadingLoopOff – LeadingLoopOn))/100L)- loop_length` Where Delta_speed is speed calculated in metre/second Leading Loop On time is the time measured in millisecond when the vehicle presence is detected. Leading Loop Off time is the time measured in millisecond when the vehicle moves out of the loop (detection zone).
Y	**Speed**	`The speed is measured by accurately measuring the time when the vehicle enters the Leading loop and when the when the vehicle enters the Trailing loop. The speed equations are given below for the dual loop classifiers. Time measurement has a resolution of 1 millisecond The measurement has rounding error for the last digit (i.e. +-1 mph or +-1kph)`  `Dspeed = (Loop_sep + loop_length)*100)/ (TrailingLoopOn – LeadingLoopOon) : m/s`  `speed = (Dspeed*36L)/100L : Km/hour`  `speed = (Dspeed*224L)/1000 : mph/hour` Where Dspeed is speed calculated in metre/second Loop Sep is the loop separation between the leading loop and the trailing loop in centimetres. Loop length is the loop length dimension of the loop respect to the flow of the vehicles in centimetres. Trailing Loop On is the time measured in millisecond when the vehicle presence is detected, Leading Loop On time is the time measured in millisecond when the vehicle presence is detected. The single loop classifier speed measurement derived from complex differential calculations and statistical analysis.
Y	**Class**	See section 3.3

Std	Field	Description
N	**Gap**	The time measured between the instant when a vehicle leaves the trailing loop and the instant when the next vehicle enters the leading loop. Gap is measured in seconds.
N	**Headway**	The time measured between the instant when a vehicle enters the leading loop and the instant when the next vehicle enters the same loop. Headway is measured in seconds.
N	**Axles**	Axles are measured by profiling the signature.
N	**Height**	A ratio that is proportional to the height of the chassis from the road surface. Actual Height = k * Height. Since k is a constant that varies from site to site, k remains an unknown. Hence height given is just proportional to the actual height.
N	**PW**	Time in milliseconds relating to the presence of the vehicle above the loop.
Y	**Direction of flow**	SP4 Leading Loop (Upstream loop) Trailing loop (Down stream loop) '+'indicates vehicle movement from leading loop to trailing loop "-" indicates vehicle movement from trailing to leading loop SP4/SLC4- Direction may be set by the user, by observing the vehicle flow. '-' is default. Single loop classifiers can't measure the direction of the travelling vehicle.
N	**Validity code**	

ASCII VBV Output

All firmware variants support two different formats of Vehicle by Vehicle output. The standard format is a shortened version of the VBV listing whilst the extended version displays all parameters including many that are not commonly used. The 'STD' field in the table above shows which parameters are included in the standard output. The extended version includes all the parameters shown above.

This basic statistical data is used to calculate further statistics on a lane-by-lane and site-by-site basis (see below).

3.3 Classification

Processed vehicles are passed through the selected classification algorithm. There are five classification algorithms, only one is run at any one instance.

Class Name	Number of Vehicle types
EURO-6	6
DOE-INI5	5
*DFT Core	16
*AUSTROADS	12

User defined	24

*DFT Core censor is issued for UK only

*AUSTROADS are issued for Australia only.

The classification are based on,

Length,

Relative chassis height

Number of axles

Axle separation

In addition a user may define their own dynamic classification tables.

3.4 Data Storage

The VBV data is stored in the compact flash card. The storage format is FAT16 file system. The compact cards must be formatted to FAT16 using any standard PC, before placing it on the unit. The unit supports two methods of data storage which are binary records or ASCII log file. The files can be read by a PC or PDA.

⚠ CAUTION

If files are read with a PDA, DO NOT Re-insert the CF back into the UNIT without reformatting it. PDA's write a temporary file to the CF card with extended naming.

The unit allows two kind of data storage.

Storage Format	File Name
Ascii	LOGDATA.TXT
Binary Records	BINDATA.DAT

The unit supports the following compact flash cards sizes.

Size	Vehicles
256Mbyte	
512Mbyte	
1024Mbyte	
2048Mbyte	

⚠ CAUTION

DO NOT INSERT ANY OTHER SIZES.

The recommended compact flash cards are Kingston and SanDisk. We do not recommend 'CRUCIAL'.

3.4.1.1 ASCII vs. Binary

When ASCII log file is chosen, the unit will store in addition to the VBV output all the Warnings or Error messages. It can also store, if selected to do so, the vehicle signatures and vehicle loop events.

When Binary is chosen only the vehicle records are stored.

Why choose ASCII?

ASCII log file should be chosen if vehicle signatures or loop events are to be recorded. It should also be chosen if a site is malfunctioning, as error and warning messages in the log file can be used to identify the cause of site malfunction.

A SCII files are slower than binary record transmission rates.

Why choose Binary?

If a site is deemed working properly then Binary storage is the preferred option. Binary vehicle records provide comprehensive vehicle details. It also allows fast traffic data transmission.

Binary record format should be chosen for remote dial-up sites where dial-up time need to be made as short as possible.

Only one of the methods is allowed.

3.4.1.2 Removing & Inserting Compact Flash Cards

Compact flash cards can be removed after closing the log file or the binary record file.

File may be closed and the compact flash card safely removed is one of the following methods

1- By pressing the front button and removing when the LED flash an X pattern.

2- From the terminal issuing closelog or vbv_save command appropriately

3- Issuing a close file command from the packet based protocol

Removing compact flash card without closing the files will result in the compact flash card file not able to read by the PC/PDA.

If the compact flash card is removed without closing the file,

1 - If the unit setup was not changed. It can be replaced in the unit and the file can be closed as above.

2 - If the unit behaviour was modified, then power off the unit, place the compact flash card and then power the unit on, close the file as above.

3.5 Data Transmission

See section 4 Data communication protocols.

3.6 Faults

To detect the faults of the unit, The following commands should be used:

3.6.1 Loop status

This command will show which of the loops are tuned and are not tuned. Those which are not tuned are either Open circuit or loop inductance is faulty,.

3.6.2 Event

Reporting an event when vehicle passes over the loop

3.6.3 Status

Shows if the CF card is detected and the size of it,

3.6.4 System variable (SVAR)

Reset

Number of Forced Resets

3.7 Communications

The classifier supports 3 physical ports.

- USB

- RS232 Modem
- RS485 Network

The USB and the Modem port provide the same message interface, with the latter defaulting to a modem layer. The modem layer of the RS232 can be turned off from USB port or device can simulate by setting the DCR level ON. The RS485 message interface is separted

The USB port baud is fixed to 57.6kbaud. Whilst the RS232 modem port switches according to modem type. Hence if it is GSM then it automatically switches to 9600, if it is US Robotics then it switch to 57.6KBaud.

*The PDA MicroDial Lite application connects via the serial port will simulate the modem connection and automatically switch to the appropriate baud rate. This ensures that irrespective of modem enabled or not PDA can always connect to the classifier.

*PC applications must connect to the module via the USB or ensure that modem is disabled.

Both ports can operate simultaneously.

The RS485 is a network communication port, running a master slave protocol. The primary function of this port is for extending the classifier using multi connectivity.

It is possible with suitable PC connectivity hardware that messages can be transferred between modules and a permanently connected PC.

3.8 Local Terminal

The classifier has a terminal interface In addition to packet protocol. Using hyper terminal or other suitable terminal software the classifier can be set-up and data monitored or retrieved using the terminal software.

The remote dial-up connection may also be established using Windows hyper-terminal.

Chapter 4 COMMUNICATION PROTOCOLS

The classifiers are made very flexible by supporting Terminal and Protocol based communication between itself and the PC/PDA. It detects the type of communication and automatically switches to the appropriate mode. This means it could switch between the protocols without any setups.

1. The Binary Packet based protocol is reliable communication, where each message has a unique id, 16-bit CRC, receive and transmit sequence numbers. Also it provide advance automatic tracking, retries and handshakes.
2. The terminal protocol is ASCII based. The user can communicate to the unit via a dump terminal such as the Windows Hyper Terminal. There are about 70 odd terminal commands.

Both protocols allows the user to

1. Setup the unit
2. Gather the status of the unit
3. View live traffic data
4. *Retrieve Stored (ASCII log)data
5. Retrieve stored (binary) data

*However if the device is storing data in Binary mode then Binary protocol must be used to retrieve stored data.

The Packet and Message formats are published in the document 'RCOM/Protocol for Profilers'.

The ASCII messages are described in the next section.

The binary packet based protocols offer reliable secure communication with the unit, whilst the ASCII based offer easy, unreliable non-secure communication.

4.1 ASCII Terminal Commands

*The HELP command lists most of the pertinent commands of each unit.

The following table provide the complete list of these commands.

1.	HELP	List all the commands
2.	VERSION	Query firmware version number
3.	DIR	Directory listing of the CF card
4.	TIME	Time query and set command

5.	DATE	Set or query date
6.	LOOPSEP	Set or query loop separation in cm
7.	LOOPLEN	Set or query loop length in cm
8.	STATUS	Print unit status
9.	LOOPSTATUS	Status of the loop
10.	ADDRESS	The units address
11.	VBV	Vehicle by Vehicle output
12.	COUNT_RESET	Reset the vehicle count ID to 0
13.	VBV_SEND_STOP	Binary packet unsolicited live traffic data.
14.	EVENT	Output loop events
15.	OPENLOG	Open an ASCII (LOGDATA.TXT) log file to store all messages, including vehicle line by line output, event output and signature outputs.
16.	CLOSELOG	Close the log file
17.	VBV_OPEN	Open a BINDATA.DAT to record vehicle records
18.	VBV_SAVE	Save BINDATA.DAT file
19.	LOG	Query log file status.
20.	ROLLING	End of file semantics.
21.	FAT_INIT	Initialise the FAT file system
22.	RECOVER	ASCII_FILE RECOVERY
23.	AUTO_RECOVER	ASCII file recovery
24.	SITE	Set or query the site name Maximum 32 characters.
25.	SITE_NO	Set or query site number
26.	CLASSIFICATION	Set or query current classification table.
27.	DMS	Data measurement and scale. Set the VBV output format as detailed or short. Set the speed measurement in MPH or KPH.
28.	STRD_MS	Straddle milliseconds
29.	LANEx_DIR	Directions filter for lane x. Only the ‘set direction’ vehicles are counted. Default is OFF Where x = 1 or 2
30.	LANEx_ADJ	Lane X speed/length adjustment factor Where x = 1 or 2
31.	BIKE_RATIO	Motorbike, small signature ratio
32.	DVDT_SENS	Differential algorithm sensitivity.
33.	DVDTW	Differential algorithm parameter
34.	LIVE	Live <loop>, real-time signature output
35.	CAPTURE_LOOP	Set the loop to capture the signature on vehicle presence.
36.	CAPTURE_NEXT	Capture and store the next vehicle
37.	SIG_TAIL	Capture signature tail-end
38.	SIG_TO_SCREEN	Filter In/Out the signature to the terminal
39.	FAST_SCAN	Loop scanning speed
40.	MODEM_TYPE	Define Modem type
41.	MODEM_INIT	Initialise the modem
42.	MODEM	Query modem status
43.	GSM_ROAMING	Set the GSM Roaming hours
44.	GSM_POWER	GSM Power timer settings

45.	SERIAL_ON	Turn ON/OFF RS232 serial lines.
46.	PRE_STATE	Set or query the preserve state interval in seconds.
47.	PRE_COUNT_RESET	Reset Prestate sector write count
48.	SVAR	Query system variables
49.	CFRD	Read CF card sector
50.	ERRORS	List of error
51.	RESET	Warm start unit. Preserved state is recovered
52.	COLDSTART	Cold start the unit. Assumes default state, Time and date must be set to enter normal mode

4.2 ASCII Terminal Command Description

The following subsection describe the terminal commands

4.2.1 Help

The Help command will provide an on-line listing of pertinent terminal commands with their listing. The following table displays the help command output to a terminal. (Please note that future versions may modify or append to this list).

```
    Application Vehicle Profiler
          Version 2.05 Date 11-07-2007
        Device Address  15
    Vehicle by Vehicle Monitoring
VBV       <VBV  1..2 OR both lanes
VBVOFF    <Turns vbv off

    Event Monitoring
EVENT     <Events  1 2, or both 3
EVENTOFF  <Events  1 2, or both 3
LOOPSTATUS   Returns ref's
LIVE  N To turn off LIVE<CR>
LOOPSEP  LANE1 LANE2 (cm))
LOOPLEN  LANE1 LANE2  (cm)
DMS  E M where E = 0 Standard VBV. M=0 Kph

     VBV Store and Binning of Data (Binary files)
BIN_STORE   Bin Save period Secs
VBV_STORE   L  L (Lane 1 2 1-on 0-off)

     Logging All screen information to text file
OPENLOG  Opens log file
CLOSELOG Turns the Logging off
ROLLING <O-OFF 1-ON> CF used as rolling buffer
PRE_STATE   Preserve State (1..3600 secs)
RESET       (Warmstart)
COLDSTART
ERRORS
STATUS
RECOVER  <sector id> <number of sectors>
AUTO_RECOVER  <0-START 1-CONTI> <No of sectors>

     FAT16 File system commands
DIR
FOPEN      <NAME OF FILE>
FCLOSE
FWRITE
FREAD
FTEXT    Text file to read
CFRD   Sector Hex Addr
     Time & Date Setup
TIME  HH:MM:SS
DATE  YYYY-MM-DD
MODEM 0 -Disable 1 Enable
MODEM_TYPE 0-US Robo 1-GSM WaveCom
GSM_ROAMING <HOURS>
```

```
SHARP_EDGE <1>  Enables detection without flux compensation
VBV Suspect  Codes. These are bit masks
 gap     # 1
 headway # 2
 speed   # 4
 axles   # 8
 length  # 16
 type    # 32
->
```

4.2.2 VERSION

The version command will display the firmware version number, and when the firmware was released. In addition it will also printout the current network address of the unit.

```
->version
VERSION 2.05  11-07-2007
Device Address  15
```

4.2.3 Dir

The DIR command gives a directory listing of files on the compact flash card. Following shows an example of directory listing.

The compact flash cards are FAT16 formatted. The size and the sector are represented in hexadecimal. Sectors are 512 bytes in length

.

```
File name          Size       Sector
BINDATA DAT     size: 00000200     sector: 0000024F
no more files
```

4.2.4 TIME

To query the current time, just type TIME

```
->TIME
TIME 11:00:20 (39619446)
```

Changing time

To change time enter TIME HH:MM:SS

The unit will echo back the time with bracket containing the time in milliseconds.

```
->TIME 11:10:15
TIME 11:10:15 (40215002)
```

Time is represented as HH:MM:SS (milliseconds since midnight).

(Note: the binary protocol allows the synchronisation of time in milliseconds).

If the unit is networked, time is automatically broadcast by the Master unit to all the slave units.

4.2.5 Date

To query the current time, just type DATE

```
->DATE
DATE 2007-08-05
```

To change date, type DATE YYYY-MM-DD

Date format is Year, Month , day each seperated by '-'

```
->DATE 2007-08-05
Committed. State-CRC=3452 Setup-CRC=7ba4
DATE 2007-08-05
```

When date is changed the unit will commit the new date to the setup. The setup data will be commited to both the default and state mirror sector.

If the unit is in default initialisation mode, changing date will automatically set it to normal mode. If the unit is networked, date is automatically broadcast by the Master unit to all the slave units.

4.2.6 Loop separation

Loop separation is the gap between the leading and trailing loop. The loop separation variable doesn't exsit for single loop classifiers or the cycle detectors.

For the dual loop classifiers loop separation must be set representing as accurately as possible the actual loop separation.

Loop separation is in ***CENTIMETRES.***

To query loop separation type LOOPSEP

```
->LOOPSEP
LOOPSEP  100 100
```

The default loop separation is in 100 cm (ie 1 m)

To change loop separation type

LOOPSEP <LANE 1> <LANE 2>

```
->LOOPSEP 150 150
Committed. State-CRC=e73b Setup-CRC=7a24
LOOPSEP  150 150
```

The above extract shows the unit loop separation changed to 150cm for lane 1 and lane 2.

When loop separation is changed the unit will commit it to the setup in the non-voltile memory. The setup data will be commited to both the default and state mirror sector.

Valid range of loop separation is 30 cm to 600 cm

```
->LOOPSEP 700 700
Out of Range ->Try (30..600 cm)Syntax error.  Try->LOOPSEP  XXX  XXX

->LOOPSEP
LOOPSEP  150 150
```

The above extract shows the loop separation being set which is out of range.

4.2.7 Loop length

Loop length is the length of loop which traverses the vehicle flow.

The length of the loop must be set representing as accurately as possible the actual loop length. For dual loop classfiers the length of the loop with in a lane pair must be the same.

Loop length is in ***CENTIMETRES.***

4.2.7.1 SP4/SLC4 Loop length

Syntax

To query loop length type

LOOPLEN

To setup loop length

LOOPLEN <LANE 1> < LANE 2 >

```
->LOOPLEN
LOOPLEN  200 200
```

The default loop length is in 200 cm (ie 2 m)

To change loop length type

```
->LOOPLEN 250 250
Committed. State-CRC=139f Setup-CRC=3e
LOOPLEN  250 250
```

The above extract shows the unit loop length changed to 250cm for lane 1 and lane 2.When loop length is changed the unit will commit it to the setup in the non-voltile memory. Valid range of loop length is 30 cm to 600 cm

```
->LOOPLEN 700 700
Out of Range ->Try (30..600 cm)Syntax error.  Try->LOOPLEN  XXX  XXX
->LOOPLEN
LOOPLEN  250 250
```

The above extract shows the loop length being set which is out of range.

4.2.7.2 SP4/SLC4 Loop length

The SP4/SLC4 has to setup loop length for each lane based on single loop.

The loop lengths are grouped as lane 1,2 and lane 3,4. The command is identical to looplen comand described as above.

Syntax

LOOPLEN_1

LOOPLEN_2

To change loop length

LOOPLEN_1 <lane1> <lane 2>

LOOPLEN_2 <lane 3><lane 4>

4.2.8 Status

Typing 'Status' provides an output detailing the unit's current operating status.

```
->status
Vehicle Profiler Ver 2.05 Release Date 11-07-2007
     Device Address  15
SITE  NOT_SET
SITE_Number 1
DATE 2007-08-05
TIME 11:51:03 (42660438)
Logging Disabled
Binary VBV File Disabled
Detected  512MByte
End_of_CF : File Closes
Signature_Capture disabled
Signature_to_Screen  enabled
Modem Enabled
Modem   WaveCom_1206B
GSM_Roaming = 1 hours
GSM Power switch timer disabled
GSM Power Auto ON in hours 120
Extended VBV with Axles & Headway &
CLASSIFICATION EURO-6
SPEED Kph
Unidir Straddle filter  Lane 1 OFF  Lane 2 OFF
Preserve Interval  20 secs
SLAVE DEVICE
```

The status shows:

- Device type (ie profiler , single loop classifier or cycle detector)
- the firmwate version and release date

- the device address.
- Site name : the default is ‘NOT_SET’
- Site number : the default is 1
- the current date
- the current time
- Text logging enabled or disabled.
- Binary VBV record saving enabled or disabled.
- The size of the compact flash card
- The sematics of how to handle when compact flash card is full (ie close it or roll over).
- Signature capture status
- Signature routing to the screen status
- Modem driver enabled for serial port status
- Type of modem currently setup (default is GSM WaveCom_1206B)
- Gsm roaming setup in hours
- GSM Power (darlingto pair) switch status
- GSM Power cycle time
- Text VBV Output format
- Current vehicle classification scheme
- Speed out measurement in KPH or MPH
- Starddle status
- State preserve interval
- Device network status – Standalone, or networked Master or Slave .

4.2.9 Loop Status

The loop status provides the loop operation status.

```
->loopstatus
Loop 0 Tuned, Ref = 2141 2141  (0)

Loop 1 Tuned, Ref = 2214 2214  (0)

Loop 2 Not Tuned
Loop 3 Not Tuned
LOOPSEP  150 150
LOOPLEN  250 250
Lane Adj 100 100
LoopScanCycles (X10)  27
LoopScan Fast (1ms)
Signature tail-end 10
T=25 C
```

```
CPU_Cycle (/ms)  39
Avg_CPU_Cycle (/ms)  38
```

The loop status shows

- Tuned status of each loop. If the loop is Open circuit or short circuit then the loops will not be tuned.
- Loop separation setup
- Loop length setup
- Lane adjustment factor setup
- Current average loop scan cycle count for every 10 millisecond
- Loop scan setup (Fast or slow).
- Signature capture – tail end add – in millisecond
- Current temparature
- Processor bandwidth: Cpu service cycle count per millisecond. Note the CPU service time = 1000/Cpu_cycle micro seconds
- Processor bandwidth : Average CP Cycle count.

4.2.10 Address

The network address and lane mapping.

4.2.11 Vehicle By Vehicle live output (VBV)

Vehicle by vehicle text output to the terminal. The VBV command allows monitoring live vehicle. The output format is dictated by the VBV format selected using the DMS (data measurment scale) command.

The purpose of the VBV command is site diagnosis and remote real time vehicle monitoring.

Syntax : VBV <n> : Vehicle by Vehicle with lane filter.

*Where n= 1 for lane 1 only , n=2 for lane 2 only and n=3 for all lanes.

VBV : VBV status query.

VBVOFF : Turn VBV live output to terminal OFF.

For SP4/SLC4 : VBV can't be filtered for lane 3 and 4. To monitor lane 3 and 4 use VBV <3>.

(*When the unit is networked as a slave , n must be set to 3).

VBV Query with VBV set to off

```
->vbv
```

```
SITE NOT_SET
SITE_Number 1
CLASSIFICATION EURO-6
SPEED Kph
VBV printing is currently set turned off - to turn it ON type VBV 3
```

VBV On with no lane filter . Extended VBV Output

```
->vbv 3
SITE NOT_SET
SITE_Number 1
CLASSIFICATION EURO-6
SPEED Mph
Length = CENTIMETER Gap & Headway=1/10sec
DATE          TIME      COUNT        LANE    LENGTH  SPEED   CLASS   GAP     HDWAY   AXLES   HEIGHT  PW      DIR
2007-08-06 10:30:38     00000007      11      304      37     2       509     512      2       71      332    -
```

Short VBV Output

```
SITE NOT_SET
SITE_Number 1
CLASSIFICATION EURO-6
SPEED Kph
Length = CENTIMETER , Gap = 1/10sec
DATE          TIME       COUNT      LANE   LENGTH      SPEED CLASS DIR
2007-08-06 10:29:40     00000005     11     401           63    2     -
```

The VBV command is primarily for outputing to the terminal live vehicle as they pass over the loop(s). Both SLP and dual loop provides the same VBV output.

VBV is automatically turned off, if VBV was requested from a remote terminal via modem, when the modem disconnects.

VBV Suspect Codes #X

If the VBV line is suspect then to each suspect line '#X' code is appended. The suspect code is a bit mask, hence it could be a combination of any or all.

Description	HEX Code
Vehicle GAP suspect	#1
Headway suspect	#2
Speed suspect	#4
Axles suspect	#8

Length Suspect	#10
Classification type suspect	#20
Straddle suspect	#40

Bit mask representation of the suspect code.

0	straddle	classtype	Length	axles	speed	headway	gap

1 : in the appropriate bit signifies the parameter is suspect.

0 : signifies the parameter is valid.

'Suspect' is defined as a parameter whose measurement either fell outside the expected values, or that the value was not expected. In the case of straddle, it signifies the vehicle was straddling both lanes.

4.2.12 VBV Count reset

Forces the VBV count to be reset to 0. The first vehicle following count reset will be '1'.

```
->count_reset
->VBV 3
SITE NOT_SET
SITE_Number 1
CLASSIFICATION EURO-6
SPEED Kph
Length = CENTIMETER , Gap = 1/10sec
DATE          TIME      COUNT        LANE    LENGTH  SPEED   CLASS   GAP     HDWAY   AXLES   HEIGHT  PW      DIR
2007-08-06 10:30:38     00000001     11      304      37     2       509     512      2       71      332    -
```

The unit will automaticall reset to 0 on overflow. Maximum count is 99,999,999. Each unit thus can count upto 100 million vehicle before overflow. The maximum vehicle record able to be stores in a 2G compact flash card is 55 million vehicles individual records.

4.2.13 VBV_SEND_STOP

The units can send unsolicited binary vehicle records. This transfer is normally setup by sending the approriate binary request packet.

Normally this transfer is started and stopped from MicroDial or suitable program equiped to handle binary protocol.

If the unit for any reason was started to send unsolicited live traffic data, but not stopped and a terminal was to be connected, this command may be issued to stop such transfer.

4.2.14 EVENT

The EVENT command send loop event output to the terminal. The event command, like the VBV command allows individual lanes to be monitored.

The purpose of the EVENT command is for site diagnosis.

Syntax : EVENT <n> : Loop event with lane filter.

*Where n= 1 for lane 1 only , n=2 for lane 2 only and n=3 for all lanes.

EVENTOFF : Turn EVENT live output to terminal OFF.

```
->event 3
Lane id idx = loop ontime offtime chassisheight
1 020 0 = 1 36206790 36207098 67 1 2  [150 30] [1 1] [1 1] [1 1]  [312 2] [1 1] [1 1] [1 1]
1 021 1 = 0 36207027 36207337 66 2 2  [128 30] [289 4] [1 1] [1 1]  [20 4] [263 9] [1 1] [1 1]
1 022 2 = 0 36210725 36211370 64 2 2  [211 26] [454 31] [1 1] [1 1]  [346 28] [589 7] [1 1] [1 1]
1 023 3 = 1 36211206 36211851 67 2 2  [225 26] [441 33] [1 1] [1 1]  [360 28] [630 4] [1 1] [1 1]
```

If the unit is set to logging mode, the events will be recorded prior to the respective VBV line.

The Event output parameters

Lane : The lane to which the loop is mapped
id : Event ID, (MOD 256)
idx : internal algorithm index
Loop ontime : Loop presence detected time in milliseconds (24 hour clock).
Loop offtime: Loop presence off time
Chassisheight: relative height

In addition to this conventional event information, each event output is tagged with the abstract format of the signature waveform. These are appended after the chassis height.

For SP4/AXD4- Overlapping events are necessary to classify a vehicle. The exception to this rule is motorbikes. Overlapping events are where leading and trailing loop 'ON' times overlap. Those events that do not overlap are rejected. Event output will specify these with an appropriate message.

For SP4/SLC4 – Single events are used to map the vehicle. This solely depends on the waveform analysis and statistical engines.
SP4/SLC4 can't set filter for lanes 3 and 4. To monitor events in these lanes
Use 'EVENT 3'.
When 'EVENT 1' is used only loop 1 event is shown, similarly for 'EVENT 2'.

4.2.15 Open Log

The command will open a logdata.txt file if no such file exists or will append to logdata.txt file if such file exits. If the unit is already logging no action will be taken.

Opening file with no log file present.- new log file will be created

```
->dir
File name          Size       Sector
no more files

->openlog
Log LOGDATA.TXT file Opening
Open  file LOGDATA TXT

Log Starting
2007-08-12 10:36:04
SITE NOT_SET
SITE_Number 1
CLASSIFICATION EURO-6
SPEED Mph
```

Open log with existing log file – the logdata.txt file will new appended with new data.

```
->dir
File name          Size       Sector
LOGDATA TXT     size: 00000400     sector: 00000210
no more files
->openlog
Log LOGDATA.TXT file Opening
Open  file LOGDATA TXT

Log Starting
2007-08-12 10:39:32
SITE NOT_SET
SITE_Number 1
CLASSIFICATION EURO-6
SPEED Mph
```

Open log when a logfile is already opened

```
->openlog
Is Logging
```

The open log will command will execute when all existing file operation is complete.

4.2.16 Close Log

The ‘closelog’ command will close the log file if logging is enabled, otherwise no action is taken.

Closelog whilst logging.

```
->closelog
```

```
->
Log Closing
2007-08-12 10:49:13     CLOSE SECTOR = 531
File Close LOGDATA TXT
Write Sector = 6 Cluster 114011956087154170887
Write Sector = 251 Cluster 114011956087154170887
```

4.2.17 VBV Binary Record file open (VBV_OPEN)

The units can be configured to save VBV file records within a BINDATA.DAT file. The unit only allows either the binary record saving or text file , both can't be done at the same time.

```
->vbv_open
->
Open  file BINDATA DAT

->dir
File name          Size       Sector
BINDATA DAT     size: 00000000     sector: 0000024F
no more files

->log
Binary VBV File Enabled
```

4.2.18 VBV Binary Record file close (VBV_SAVE)

If a BINDATA.DAT file is opened, the file will be closed.

```
->vbv_save
->log
Logging Disabled
Binary VBV File Disabled
->dir
File name          Size       Sector
BINDATA DAT     size: 00000000     sector: 0000024F
no more files
```

4.2.19 Query log/saving status (Log)

The log status can be queried by typing log.

Logging is enabled, ie LOGDATA.TXT file open.

```
->log
Logging Enabled
```

VBV Record saving eanbled, ie BINDATA.DAT file open.

```
->log
Binary VBV File Enabled
```

No data is being saved in the CF card. Ie both LOGDATA.TXT and BINDATA.DAT closed.

```
->log
Logging Disabled
Binary VBV File Disabled
```

4.2.20 Rolling

The 'Rolling' allows how to setup the end of file (due to memory full) is handled

If Rolling is set, when the memory is full the file is restarted from the start. If Rolling is not set the file is closed and no further logging is carried out.

To query

```
->rolling
End_of_CF : File Closes
```

To set to rolling (on)

```
->rolling 1
Committed. State-CRC=7c4b Setup-CRC=d93d
rEnd_of_CF : Rolling Buffer
```

To reset rolling (off)

```
->rolling 0
Committed. State-CRC=f786 Setup-CRC=8687
End_of_CF : File Closes
```

4.2.21 Initialise the fat (FAT_INIT)

Forced initialise the FAT area. If FAT is already initialised no action is taken. Otherwise FAT area is initialised. The manual use of this command is not necessary, as FAT area is automatically intialised when the unit detects CF card.

4.2.22 RECOVER

This command is used to recover the stored data in the Compact flash card sector by sector.

The command will ouput to the screen all the data from the starting sector to the end sector. The output is stopped when end sector is reached or when key is pressed. If the sector count is set , it will stop when the number of sectors printed equals to the sector count.

Syntax:

Recover

Recover <sector id>

Recover <sector id> <sector count>

Where

sector id is in hex

sector count is decimal

Starting sector of a file is displayed in the directory table.

```
->dir
File name          Size       Sector
LOGDATA TXT     size: 00000200     sector: 00000210
BINDATA DAT     size: 00000000     sector: 00000218
no more files
```

For different CF card sizes starting sector is different.

The recover command without a paramters will automatically point to the FAT area where the first file starts.

```
->recover
->
Log Starting
2007-10-03 14:57:07
SITE NOT_SET
SITE_Number 1
CLASSIFICATION EURO-6
SPEED Kph
Committed. State-CRC=296c Setup-CRC=8687
.
.
.
Recover Last sector id = 532
```

Recover with the sector will start printing from the sector. In the below example the sector equals to start of the logdata.txt file (see dir output above).

```
->recover 210
T=28 C
->
Log Starting
2007-10-03 14:57:07
```

```
SITE NOT_SET
SITE_Number 1
CLASSIFICATION EURO-6
SPEED Kph
Committed. State-CRC=296c Setup-CRC=8687
.
.
.
Recover Last sector id = 533
```

Recover sector followed by the count of sectors to printout.

```
->recover 210 1
->
Log Starting
2007-10-03 14:57:07
SITE NOT_SET
SITE_Number 1
CLASSIFICATION EURO-6
SPEED Kph
Committed. State-CRC=296c Setup-CRC=8687
.
.
.

 Completed CF Scan. Last Sector = 529
```

4.2.23 AUTO_RECOVER

Auto_recover command will automatically point to the start of the sector where the first file starts. The initialise option must be set to 0. To continue from the next sector set the continue option 1.

For each command it will countdown the number of sectors to print and then stop. If any key is pressed the output will be stopped.

Syntax

Auto_recover <initialise/continue> < number of sectors to print>

Where

initialise/continue : 0 – intilaise, 1- continue

no sectors : range 1 to 65535

4.2.24 SITE

Site defines the site name. This is an alphanumerical character between 1 and 30. When set from terminal the space character is not allowed. However when set through MicroDial via the binary packet space character is allowed.

Syntax

Site <site_name>

Default setup : NOT_SET

```
->site
SITE  NOT_SET
->site test_site
Committed. State-CRC=5288 Setup-CRC=ff43
SITE  test_site
```

4.2.25 SITE_NO

Site number is assigned by the user.

Syntax

Site_no <n>

Where n = 0.. 32767

Default setup : 1

```
->site_no
SITE_Number 1
->site_no 9
Committed. State-CRC=b18f Setup-CRC=35d0
SITE_Number 9
```

4.2.26 CLASSIFICATION

Set the classification table to use. There are 5 settings for classifications.

Syntax

Classification < id >

Where classification 'id' is a number 0.. 5.

ID Number	Classification
0	No classification
1	EUR0-6
2	DOE-NI 5
3	DTP-CORE
4	*AUSTOROADS
5	USER Defined

```
->classification
CLASSIFICATION EURO-6
->classification 0
Committed. State-CRC=4d58 Setup-CRC=c93e
CLASSIFICATION None
```

```
->classification 1
Committed. State-CRC=e597 Setup-CRC=eba6
CLASSIFICATION EURO-6
->classification 2
Committed. State-CRC=13e5 Setup-CRC=9c06
CLASSIFICATION DOE-NI-5
->classification 3
Committed. State-CRC=7db0 Setup-CRC=be9e
CLASSIFICATION DTP-CORE
->classification 4
Committed. State-CRC=876b Setup-CRC=634e
CLASSIFICATION AUSTROADS
->
T=27 Cclassification 5
Sorry no user defined table found
->classification 6
Out of range 0..4
```

4.2.27 DMS

The DMS allows data measurement and VBV output format

Syntax DMS <vbv fmt> < speed measurement>

Where

Vbv fmt = 0 : Standard short VBV is output or logged

Vbv fmt = 1: Extended VBV is output or logged

Speed Measurement = 0 : Speed measured in Kph

Speed Measurement = 1 : Speed measured in Mph

```
->dms
Extended VBV with Axles & Headway &Speed Measured in Kph
->dms 0 0
Committed. State-CRC=cbb5 Setup-CRC=eb4c
Std VBV & Speed Measured in Kph
->dms 0 1
Committed. State-CRC=69c4 Setup-CRC=af47
Std VBV & Speed Measured in Mph
->dms 1 1
Committed. State-CRC=ef24 Setup-CRC=2745
Extended VBV with Axles & Headway &Speed Measured in Mph
->dms 1 0
Committed. State-CRC=dc1c Setup-CRC=634e
Extended VBV with Axles & Headway &Speed Measured in Kph
```

4.2.28 STRD_MS

Straddle algoritham time. The straddle algoritham uses straddle loop pairs. The straddle loops pairs are those adjacent to the next lane. The setting of straddle pairs are loop 1 and 3 and loop 2 and 4. If vehicles straddle both lane then only one lane is counted.

Syntax

Strd_ms < n>

Where n is in milliseconds, range 0..254.

(If n = 0 straddle algoritham is turned off.)

The default setup is 50 milliseconds.

```
->strd_ms
Straddle Ms 50
->strd_ms 100
Committed. State-CRC=9321 Setup-CRC=44c9
Straddle Ms 100
->strd_ms 255
Straddle Ms 100
->strd_ms 254
Committed. State-CRC=da9f Setup-CRC=692d
Straddle Ms 254
```

4.2.29 LANEX_DIR (SP4Only)

Unidirectional straddle filter setup. This filter allows counting vehicle on a specified direction only.

Syntax

LANE1_DIR n

LANE2_DIR n

Where n = 0,1,2

If n = 0 then straddle filter is off.

Direction values assumes either '+' or '-'

If n = 1 then straddle filter is setup to count only when loops turn on loop 1 then loop 2

Direction value '+'

If n = 2 then straddle filter is setup to count only when loops turn on loop 2 then loop 1

Directiob value '-'

Default straddle filter is OFF. (ie n = 0)

```
->LANE1_DIR
Unidir Straddle filter  Lane 1 OFF
Where '+' is loops 1 ON then 2 ON
->LANE1_DIR 1
Committed. State-CRC=e28d Setup-CRC=2f62
Unidir Straddle filter  Lane 1 '+'
Where '+' is loops 1 ON then 2 ON
->LANE1_DIR 2
Committed. State-CRC=127a Setup-CRC=b529
Unidir Straddle filter  Lane 1 '-'
```

```
Where '+' is loops 1 ON then 2 ON
->LANE1_DIR 0
Committed. State-CRC=4446 Setup-CRC=692d
Unidir Straddle filter  Lane 1 OFF
Where '+' is loops 1 ON then 2 ON
```

4.2.30 LANEX_ADJ

Adjust the length and speed by a fixed percentage.

Syntax

LANE1_ADJ <n>

LANE2_ADJ <n>

Where n = 50% to 200%

```
->lane1_adj
Lane 1 Adj 100
->lane2_adj
Lane 2 Adj 100

->lane1_adj 50
Committed. State-CRC=22cd Setup-CRC=2722
Lane 1 Adj 50
->lane2_adj 40
Out of Range ->Try (50..200)
Lane 2 Adj 100
```

4.2.31 BIKE_RATIO12 (SLP Only)

Sets bike ratio to lane 1 and 2

4.2.32 BIKE_RATIO34 (SLP Only)

Sets bike ratio to lane 3 and 4

4.2.33 BIKE_RATIO (SP4/AXD4 Only)

Motobikes exhibit very low profile. In most cases motorbikes will not result in overlapping events, ie presence of the vehicle on both loops overlapping at some point of time.

Motobikes are distinguished from the low chasis profile plus the length of the vehicle.

Bike ratio defines the chassis height ratio, above which should be considered for bikes when none overlapping events occur.

Classification of motobikes is made :

If the length is below 3 m and the chasis profile is above bike_ratio.

Syntax

Bike_ratio <lane 1> <lane 2>

Defaul setting is 120 for both lanes.

```
->bike_ratio
Bike Height Ratios  120 120
->bike_ratio 300 300
Committed. State-CRC=9f08 Setup-CRC=63d8
Bike Height Ratios  300 300
```

4.2.34 DVDT

The differential algoritham profile the waveform on realtime. The sensitivity setting allows to tune the profiling algoritham,

Syntax

Dvdt <n>

Where n = 1 to 10

1 being high sensitivity and 10 being the lowest sensitivity.

The default is 3.

Care should be exercised when changing this setting.

```
->dvdt
Differential Sensitivity 3
->dvdt 11
Out of Range -> Try   (1..10)
Differential Sensitivity 1
->dvdt 10
Committed. State-CRC=a5fd Setup-CRC=43fd
Differential Sensitivity 10
->dvdt 3
Committed. State-CRC=31c0 Setup-CRC=692d
Differential Sensitivity 3
```

4.2.35 DVDTW

This setting covultion setting for the differential algoritham.

Syntax

Dvdtw <n>

Where n = 5 to 50

```
->dvdtw
Dvdt win  50
->dvdtw 0
Dvdt win  50
```

```
->dvdtw 10
Committed. State-CRC=cd72 Setup-CRC=61c6
Dvdt win  10
->dvdt2_w 50
Committed. State-CRC=7ceb Setup-CRC=692d
Dvdt win  50
```

4.2.36 LIVE – Continuos signature output

The signature of any of the loop can be printed out on the serial port, or logged in the compact flash card on realtime.

Syntax

Live < loop number>

Live

Where : loop number is 1 to 4.

Live without any parameter turns off the live output.

```
->live
Usage LIVE  <N> n= 1 to 4. Turn off LIVE<cr>
->live 1
->2214 2215 2215 2215 2214 2215 2215 2215 2214 2215 2214 2215 2216 2215
2216 2215 2215 2216 2216 2215 2216 2215 2215 2214 2215 2214 2215 2215 2214
2214 2214 221
```

Live output will result in about 1000 signature per second. The continuos signature is output irrespective of vehicle events.

4.2.37 Capture loop – presence event triggered signature output

Capture loop will capture signature when vehicles are present.

Syntax

Capture_loop <n>

Where n = 0.. 4, 0 - capture loop is turned off. 1..4 loop number

Default setting is capture disabled

The following shows a trace where both event and VBV is turned on with capture loop set to capture event in loop 1. Signature tail is set to 20.

```
->capture_loop
Signature_Capture disabled
->capture_loop 1
Signature_Capture for loop 1
->event 1
```

```
Lane id idx = loop ontime offtime chassisheight
->vbv 3
SITE 1234
SITE_Number 32767
CLASSIFICATION AUSTROADS
SPEED Kph
Length = CENTIMETER Gap & Headway=1/10sec
DATE          TIME      COUNT       LANE  LENGTH      SPEED CLASS GAP
      HDWAY AXLES HEIGHT      PW    DIR
->2207 2207 2207 2208 2207 2206 2206 2206 2206 2205 2205 2205 2204 2203
2203 2203 2203 2203 2202 2201 2201 2200 2200 2200 2199 2199 2198 2197 2197
2196 2196 2196 2194 2194 2194 2193 2191 2191 2190 2190 2190 2189 2189 2189
2188 2187 2185 2185 2185 2185 2184 2185 2183 2183 2183 2183 2184 2183 2184
2183 2183 2184 2183 2184 2185 2184 2184 2184 2185 2184 2184 2183 2184 2183
2183 2183 2182 2183 2183 2183 2183 2183 2182 2183 2181 2182 2182 2181 2181
2181 2180 2180 2179 2179 2178 2178 2178 2177 2176 2177 2177 2177 2176 2177
2177 2177 2177 2179 2179 2180 2180 2181 2183 2183 2184 2185 2186 2187 2187
2188 2189 2189 2191 2192 2192 2193 2194 2196 2196 2197 2198 2198 2199 2199
2201 2201 2201 2202 2203 2205 2205 2206 2205 2206 2207 2207 2208 2208 2208
2209 2209 2210 2209 2209 2211 2211 2212 2211 2212 2212

1 001 1 = 0 75995216 75995714 63 2 2  [207 29] [358 35] [1 1] [1 1]  [327
34] [507 2] [1 1] [1 1]

2212 2213 2212 2212 2212 2213 2211 2212 2211 2212 2211 2213 2213 2212 2211
2212 2212 2211 2213 2212 2212 2212 2212 2211 2211 2211 2212 2213 2213 2213
2213 2213 2213 2212 2213 2212 2213 2214 2212 2212 2213 2213 2213 2213 2213
2213 2213 2212 2212 2214 2214 2213 2212 2214 2212 2213 2214 2214 2214 2212
2213 2214 2213 2213 2214 2213 2214 2214 2214 2213 2213 2213 2213 2214 2213
2213 2213 2213 2213 2213 2215 2213 2213 2213 2214 2214 2213 2214 2213 2213
2213 2213 2213 2214 2213 2213 2212 2213 2213 2213 2214 2213 2214 2213 2213
2214 2214 2214 2213 2213
1 002 2 = 1 75995613 75996254 63 2 2  [168 28] [408 33] [1 1] [1 1]  [348
33] [586 6] [1 1] [1 1]

2007-10-10 21:06:37      00000101   1     173    27   1       1     1    2
       63    498  +  <#3>
->capture_loop 0
Signature_Capture disabled
```

4.2.38 Capture Next – presence triggered single event signature output

Capture next will record the signature of the next vehicle within the cf card, indicating the position where the capture was stored. The capture command is automatically disabled at the end of the reording.

Syntax

Capture_next <n> where n=1 to 4 for the loop

Recorded signature may be printed using recover comand.

Please note that this

command will disable the sig_to_screen.

Log must be enabled.

The default setting is disabled.

Following trace shows capture of loop 1. The signature is then replayed using the recover command

```
->capture_next 1
Capture_Start_Sector 1 690

->2208 2207 2208 2207 2206 2205 2205 2203 2203 2203 2202 2202 2201 2201
2199 2199 2198 2197 2196 2195 2195 2193 2192 2192 2191 2191 2189 2188 2188
2188 2186 2184 2185 2183 2182 2182 2181 2182 2180 2181 2180 2180 2179 2180
2179 2181 2181 2180 2180 2181 2181 2181 2182 2181 2181 2182 2181 2183 2182
2182 2183 2182 2182 2182 2183 2183 2183 2183 2182 2183 2183 2183 2182 2182
2183 2183 2182 2184 2183 2184
1 003 3 = 1 76196887 76197574 63 1 2  [274 35] [1 1] [1 1] [1 1]  [610 2]
[1 1] [1 1] [1 1]
2183 2183 2183 2182 2183 2183 2182 2182 2183 2182 2185 2186 2186 2188 2187
2189 2189 2190 2192 2192 2193 2194 2194 2194 2196 2197 2198 2198 2199 2199
2200 2202 2202 2202 2203 2204 2204 2204 2204 2204 2206 2207 2206 2206 2207
2207 2208 2209 2209 2209 2208 2209 2210 2210 2211 2210 2212 2211 2211 2212
2212 2212 2212 2213 2212 2212 2213 2211 2212 2212 2213 2211 2212 2213 2213
2212 2213 2211 2212 2212 2211 2211 2212 2213 2211 2212 2212 2213 2213 2214

1 004 4 = 0 76197351 76197810 67 1 2  [169 33] [1 1] [1 1] [1 1]  [477 1]
[1 1] [1 1] [1 1]
2212 2212 2212 2213 2213 2213 2213 2213 2212 2213

2007-10-10 21:09:59      00000102   1     239    23   1     2013  2019   1
       63    687  -
2213 2214 2213 2214 2213 2213 2212 2213 2213 2213 2214 2213 2213 2214 2213
2213 2213 2214 2214 2214 2213 2214 2213 2212 2213 2213 2214 2212 2213 2214
2213 2213 2214 2213 2213 2214 2213 2212 2213 2212 2214 2213 2214 2213 2212
2212 2214
Completed CF Scan. Last Sector = 693


->recover 2b2 3
->capture_next 1
Capture_Start_Sector 1 690

->2208 2207 2208 2207 2206 2205 2205 2203 2203 2203 2202 2202 2201 2201
2199 2199 2198 2197 2196 2195 2195 2193 2192 2192 2191 2191 2189 2188 2188
2188 2186 2184 2185 2183 2182 2182 2181 2182 2180 2181 2180 2180 2179 2180
2179 2181 2181 2180 2180 2181 2181 2181 2182 2181 2181 2182 2181 2183 2182
2182 2183 2182 2182 2182 2183 2183 2183 2183 2182 2183 2183 2183 2182 2182
2183 2183 2182 2184 2183 2184

1 003 3 = 1 76196887 76197574 63 1 2  [274 35] [1 1] [1 1] [1 1]  [610 2]
[1 1] [1 1] [1 1]
```

```
2183 2183 2183 2182 2183 2183 2182 2182 2183 2182 2185 2186 2186 2188 2187
2189 2189 2190 2192 2192 2193 2194 2194 2194 2196 2197 2198 2198 2199 2199
2200 2202 2202 2202 2203 2204 2204 2204 2204 2204 2206 2207 2206 2206 2207
2207 2208 2209 2209 2209 2208 2209 2210 2210 2211 2210 2212 2211 2211 2212
2212 2212 2212 2213 2212 2212 2213 2211 2212 2212 2213 2211 2212 2213 2213
2212 2213 2211 2212 2212 2211 2211 2212 2213 2211 2212 2212 2213 2213 2214

1 004 4 = 0 76197351 76197810 67 1 2  [169 33] [1 1] [1 1] [1 1]  [477 1]
[1 1] [1 1] [1 1]
2212 2212 2212 2213 2213 2213 2213 2213 2212 2213

2007-10-10 21:09:59      00000102   1     239    23   1     2013  2019   1
       63    687  -
2213 2214 2213 2214 2213 2213 2212 2213 2213 2213 2214 2213 2213 2214 2213
2213 2213 2214 2214 2214 2213 2214 2213 2212 2213 2213 2214 2212 2213 2214
2213 2213 2214 2213 2213 2214 2213 2212 2213 2212 2214 2213 2214 2213 2212
2212 2214 221
 Completed CF Scan. Last Sector = 693
->
```

4.2.39 Signature tail (SIG_TAIL)

Capture signature, captures the vehicle signature on vehicle presence. The tail end of the signature may be extended beyond the presence time by setting the signature tail in milisecond.

Syntax

Sig_tail <n>

Where n = 1 .. 254 in milliseconds.

Default value is 10.

```
->sig_tail
Signature tail-end 10
->sig_tai_l 255
Out of Range -> Try   (1..254)
Signature tail-end 10
->sig_tail 20
Committed. State-CRC=5445 Setup-CRC=5365
Signature tail-end 20
```

4.2.40 SIG_TO_SCREEN

The signature output is send both the file write (logging) process and the terminal output (screen) process.

The output to the screen can be enabled or disabled using this command without effecting the logging process.

MicroDial would normally parse the terminal characters and appropriately route them to the graphical window and signature byte. However a dump terminal will just clog up the screen with lsignature data.

Syntax

Sig_to_screen <n>

Where n = 0 : screen output is disabled

n = 1: Screen output is enabled

Default signature to screen is enabled.

```
->sig_to_screen 0
Signature_to_Screen disabled
->sig_to_screen 1
Signature_to_Screen  enabled
```

4.2.41 FAST_SCAN

Loop scan sampling frequency can be set to fast or slow using this command.

Syntax

Fast_scan <n>

Where n = 0 : slow loop scan

n = 1 : fast loop scan

Default is Fast scan.

```
->fast_scan
LoopScan Fast (1ms)
->fast_scan 0
Committed. State-CRC=6d2e Setup-CRC=c430
LoopScan Slow (2ms)
->fast_scan 1
Committed. State-CRC=1317 Setup-CRC=5365
LoopScan Fast (1ms)
```

4.2.42 MODEM_TYPE

This command allows to select the internal modem driver.

Syntax

Modem_type < modem_id >

Modem Id	Description	Settings
1	GSM Wavecom modem 9600.N.8.1	9600,n,8,1 Full hardware handshake
2	US Robotics 57.6k	57600,n,8,1 Full hardware handshake

The default modem type is WaveCom 1206b.

```
->modem_type
Committed. State-CRC=58a4 Setup-CRC=5365
```

```
Modem   WaveCom_1206B
->modem_type 2
Committed. State-CRC=934a Setup-CRC=9399
Modem  US_Robotics
->modem_type 1
Committed. State-CRC=4478 Setup-CRC=5365
Modem   WaveCom_1206B
```

4.2.43 MODEM_INIT

The command will initialise the modem. This command may be used if a modem is hot swapped. Modems are initialised on reset or when modem_type command is issued.

Syntax

Modem_init

```
->modem_init
```

4.2.44 MODEM

Modem command can be used enable or disable the modem drivers. If the modem is disabled then the serial port may be used by a dump terminal (ie hyper terminal).

Syntax

Modem <n>

Modem

Where n = 0 disables the modem driver

n = 1 enables the modem driver

Default modem is enabled.

```
->modem 0
Modem Disabled
->modem 1 1
Modem Enabled
```

4.2.45 GSM_ROAMING

GSM Roaming command allows to set the GSM roaming time varable and enable or disable the roaming algoritham. This allows the GSM modem driver to issue GSM network initialisation command.

GSM roaming is only valid if GSM modem type is selected.

Syntax

Gsm_roaming <n>

Gsm_roaming

Where n = 0.. 255

If n = 0 then GSM roaming is disabled, else time is set in hours.

Default roaming is enabled and set to hourly roaming.

```
->gsm_roaming
GSM_Roaming = 1 hours
->
T=35 C
->gsm_roaming 0
Committed. State-CRC=5e3b Setup-CRC=6a68
GSM_Roaming disabled
->gsm_roaming 25
Committed. State-CRC=30cc Setup-CRC=bdf
GSM_Roaming = 25 hours
->gsm_roaming 2
T=35 C56
Committed. State-CRC=72fc Setup-CRC=6a68
GSM_Roaming disabled
->gsm_roaming 257
Committed. State-CRC=16cc Setup-CRC=5365
GSM_Roaming = 1 hours
```

4.2.46 GSM_POWER

To turn off gsm
gsm_power 0
To turn GSM on permanently
Gsm_power 40

To turn on gsm on timer

gsm_power <hour> <minutes on> hour is 0..23 MINUTES is 1.. 1439

```
->gsm_power
GSM Power switch timer disabled
->
->gsm_power 24 0
Syntax Error
To turn on:  GSM_POWER <HOUR> <MINS ON> where hour 0..23. Min 0..1440
To turn off: GSM_POWER 0
->gsm_power 23 10
Committed. State-CRC=5687 Setup-CRC=c19d
Daily GSM Power up on the hour 23 For 10 mins
->gsm_power 0
Committed. State-CRC=4723 Setup-CRC=5365
GSM Power switch timer disabled
->gsm_power 30 10
Committed. State-CRC=d40 Setup-CRC=ac93
Hourly GSM Power up For 10 mins
->gsm_power 0
```

```
Committed. State-CRC=e1af Setup-CRC=5365
GSM Power switch timer disabled
```

4.2.47 SERIAL_ON

The serial port can be turned on or off. Turning the serial port will save power.

Syntax

Serial_on <>n>

Where n = 1 serial port enabled

n = 0 serial port disabled

Default is serial port enabled.

```
->serial_on 11
->serial_on 00
```

4.2.48 PRE_SATE

The state of the unit is preserved in the non-voltile memory. The state is refered to as the operational parameter data values. Some of these data values (states) are fixed whilst others are dynamic.

Syntax

Pre_state

Pre_state <n>

Where n = 1 to 3600 in seconds.

Example

Loop length , Loop separation and many of the above user setups, assume a fixed state.

Time, date, write sector of the file, vehicle count, file open state , file fat area etc are dynamic states

The unit stores both fixed and dynamic states into the non-voltile flash (inside the compact flash card).

If the unit went through a power failture, it recovers the state. This allows the unit to be impervious to any power failures. Simply putting after power is restored it operates as normal as if no power failure occured.

Pre_state allows the user to setup the time in seconds to save the state. This is a variable from 1 second to 3600. The deafult is 30 seconds.

```
->pre_state
Preserve Interval  30 secs
```

```
->pre_state 3601
Syntax error.  Try->PRE_STATE   Interval (1..3600 secs)
->pre_state 3600
Committed. State-CRC=be3 Setup-CRC=5948
Preserve Interval  3600 secs
->pre_state 30
Committed. State-CRC=54ba Setup-CRC=5365
Preserve Interval  30 secs
```

4.2.49 PRE_COUNT_RESET

A count is kept on the number of times the data is stores in the dynamic sector where the state is stored. This count is used to keep track of how many times a sector has been written. If it exceeds 500,000 writes the next sector is used.

This allows the unit to use sectors within the CF card conforming to the the write cycle specification.

To inspect the current write cycle examine the output from the SVAR

```
Pre-State [Sect 499985. Writes 57718]
```

The first parameter is the current sector used and the second varible is the number of time that sector is being written.

The unit will automatically read the sector count and correct the count value. User is not expected to use this command.

If the compact flash card was changed without informing the unit (ie press the front button or from reset) then user may reset the count using this command.

Syntax

Pre_count_reset

```
->svar
Pre-State [Sect 1000209. Writes 7018]
```

4.2.50 ystem variables (SVAR)

This command lists all the system variable states.

```
->svar
Pre-State [Sect 499985. Writes 57718]
Critical Interval 20
SetupCRC 20e2
StateCRC = 9761
Resets = 0
Forced Resets = 0
Fault Code 20
```

```
ExtVBV  1
Classification 0
Rolling Buffer 0
ModemType 1
Roaming Hour 1
Differential Sens 3
Active Port 1
File Open  0
Log Open   0
ReadOnly   0
VBV Open   0
Tx Sector 528 528 64
Subsector/Offset  63535623675449ý 114011879121340208740
No of records 1
VBV Count 30264191979125500800
Num_Fats = 2
Fat_Start[16273871783090162Ã] = 6
Fat_Start[1] = 251
Fat_Start[2] = 114011857131107648007
Fat_Start[3] = 114011857131107648007
P_Start = 0xa479e3926000039§
Secs_Cluster  = 0x8
Bytes_Sector = 0x200
Secs_Fat = 0xf5
Root_Start = 0x1f0
Res_Secs = 6
File_Name  LOGDATA TXT
start_cluster = 0x2
start_sector =  0x210
Fat_OpenCluster= 0x2
file_number = 0
Fat_OpenSec= 0x213
end_sector = 0x214
file_size = 0x800
file_size_start = 0x600
LOGDATA = 1
BINDATA = 0
->
```

The system variables is primarily used in case of site debugging is required.

4.2.51 Read sector CFRD

Compact Flash card read sector in hex output.

Syntax

CFRD <XX>

Where

xx – is the hex value of the sector.

```
->cfrd 24f<591>
0a 0d 4c 6f 67 20 53 74  61 72 74 69 6e 67 0a 0d    ..Log Starting..
```

```
32 30 30 37 2d 31 30 2d  30 38 20 31 31 3a 31 37    2007-10-08 11:17
3a 31 31 09 0a 0d 53 49  54 45 20 4e 4f 54 5f 53    :11...SITE NOT_S
45 54 0a 0d 53 49 54 45  5f 4e 75 6d 62 65 72 20    ET..SITE_Number
31 0a 0d 43 4c 41 53 53  49 46 49 43 41 54 49 4f    1..CLASSIFICATIO
```

4.2.52 Errors

List of errors counts and traps. If there are no errors then no output is made. This is used mainly for debugging.

4.2.53 Reset

Reset the unit similar to pressing the reset button. Reset doesn't change the state of the units but continues normally,

Reset procedure

- last state is recovered

- Loop oscillator is restarted

In essence reset just continues the operation from the point of instance of the last state saved state,

```
->reset
Recovering State
Ok Sectors Mirror 1000208  Pre 1000209
DynSector 1000209 crc f414
State Recovered [1000209] State-CRC f414 Setup CRC 5365
 Loop Signature Analysis Module.
 Terminal Mode
```

4.2.54 Cold Start

Coldstart will reset the unit from a default state. Cold start assumes the same state if the unit was powered up with no compact flash card or a totally new compact flash card.

Both date and time will be reset. Time will start at 0 and date will be set to 2006-01-01.

When the unit is cold started, the led's will flash continuosly. The unit will continuosly check the Compact flash card. If a CF with setup is inserted then that setup will be assumed.

The terminal will printout the message "Checking CF...."

The following shows the coldstart being issued. The unit will assume default state.

```
->coldstart
Recovering State
ATS0=2=0000F CRC 7a8c 9e3941000039e8007 . Default State AT
->
Checking CF...
Checking CF...
->status
Vehicle Profiler Ver 2.05 Release Date 11-07-2007
    Device Address  15
SITE  NOT_SET
SITE_Number 1
DATE 2006-01-01
TIME  00:01:58 (117526)
Logging Disabled
Binary VBV File Disabled
End_of_CF : File Closes
Signature_Capture disabled
Signature_to_Screen  enabled
Modem Enabled
Modem   WaveCom_1206B
GSM_Roaming = 1 hours
GSM Power switch timer disabled
GSM Power Auto ON in hours 120
Extended VBV with Axles & Headway &
CLASSIFICATION EURO-6
SPEED Kph
Unidir Straddle filter  Lane 1 OFF  Lane 2 OFF
Preserve Interval  20 secs
modsetup_count 114011989072503011847
SLAVE DEVICE
->
```

If the user changes any of the system parameters the unit will automatically switch from the coldstart state.

Example if the date is changed then the unit will go into normal state

```
->date 2007-09-18
Committed. State-CRC=7d29 Setup-CRC=7ba4
DATE 2007-09-18
```

Whilst in ‘default’ mode the unit will not save the state.

Chapter 5 BINARY PACKET/MESSAGE PROTOCOL

5.1 Scope

This section provides a brief description of the binary packet protocol and the related objects.

This chapter doesn't describe the parameters within the objects.

The following sections limits the description to the protocol, both message and data link layer and also the message objects and the composition of the message objects.

The objects described are limited to those which are transmittable between the unit and the peer PC. It does not describe other objects that are internal to the unit.

5.2 Data link layer (RACCOM packet protocol)

5.2.1 Frame format:

<start flag - 1 byte ><address 1-byte ><ctrl -byte ><tx_seq_no -1 byte ><rx seq no -1byte ><length - 2 BYTE ><MESSAGE - VARIABLE><CRC CCITT 16 -2BYTES>

Where

START FLAG :

START_FLAG 0x7E

DLE 0X7

5.2.2 Byte stuffing algoritham

Byte stuffing is implemented. This ensures that start flag is allowed in the packet itself.

<DLE><STARTFLAG>

<DLE><DLE>

DLE character is removed on each of above occasions. Address character is fixed for the device is set to 0x2

Control characters are

PKT_INFO = 0x20

PAKT_POLL = 0x21

5.2.3 Transmit and Receive Sequence numbers

Sequence modulo is implemented. Currently this is 128.

This defines the window of frames. Ack is inherent with the sequence from the peer device.

5.2.4 Cyclic Redundancy Check

CRC is carried out on the data message only.

CRC calculation strictly comply with 16 bit FCS (CRC -16) as defined by IS0 3309.

(Please note from versions 2.0 of SP4 and 1.17 SP4/SLC4 onwards strictly comply with IS03309).

5.2.5 ACK and NAK frames

Frame layer handshake protocol is currently not implemented due to the implementation of such protocol in the Message layer i.e. REQ-REPLY-ACK pardigm.

5.3 MESSAGE Layer

5.3.1 Message description

```
The 'MESSAGE' part of the packet is passed to the application layer.
The message format  described below


message =
      <ADDR 1 byte><LENGTH 2 bytes><MESSAGE ID - 4bytes><DATA - VARIABLE>

where
      Length =  MESSAGE ID + DATA  (ie 4 + data length)

      Maximum message size = 32768


Message ID's are unique, describes an object
```

5.3.2 Object Identifier (OID) crossreference table

5.3.3 System wide object details

Object ID	Description	Value
SysOID.ModSetup_OID	Container	121
SysOID.ModSetup_common_OID	InnerClass	124
SysOID.VbvBins_OID	Container	145
SysOID.VbvBins_vbv_OID	InnerClass	148
SysOID.FileData_OID	Container	238
SysOID.FileData_raw_OID	InnerClass	241
SysOID.ClassTable_OID	Container	247
SysOID.ClassTable_table_OID	InnerClass	250
SysOID.TrafficData_OID	Container	253
SysOID.TrafficData_td_OID	InnerClass	256
SysOID.ModSetup_dynamic_OID	InnerClass	259

5.3.4 SP4 specific object ID's

Object ID	Description	Value
AppOID.ModuleState_OID	Container	5024
AppOID.ModuleState_sys_OID	InnerClass	5027
AppOID.ModuleState_file_OID	InnerClass	5030
AppOID.ModuleState_fat_OID	InnerClass	5033

5.3.5 SLP specific objec idt details

Object ID	Description	Value
AppOID.SlpState_OID	Container	5082
AppOID.SlpState_sys_OID	InnerClass	5085
AppOID.SlpState_file_OID	InnerClass	5088

AppOID.SlpState_fat_OID	InnerClass	5091

5.3.6 System wide object composition

The following section defines the transmittable objects. The object composition below is for illustrative use only; it does not show the structure length of classes or variable parameters.

For the C structure please contact R.T.E.M Ltd.

5.3.6.1 ClassTable

```
class ClassTable
{
     table  table_obj  =  new  table();
     class  table
     {
               //Fixed size = false
                static final int OID = SysOID.ClassTable_table_OID;
                short  validCode;
                byte  class_id;
                short[ ]  inter_gap1;
                short[ ]  inter_gap2;
                short[ ]  inter_gap3;
                short[ ]  inter_gap4;
                short[ ]  min_length;
                short[ ]  max_length;
                byte[ ]  chasisFactor;
                byte[ ]  axles;
                byte[ ]  axleGapTol;
                byte  use_axlegaps;
     }
}
```

5.3.6.2 FileData

```
class FileData
{
     raw  raw_obj  =  new  raw();
     class  raw
     {
               //Fixed size = false
                static final int OID = SysOID.FileData_raw_OID;
                byte[ ]  bytebuf;
                int  sector_id;
                byte  subSector;
     }
}
```

5.3.6.3 ModSetup

```
class ModSetup
{
     common  common_obj  =  new  common();
     class  common
     {
               //Fixed size = false
                static final int OID = SysOID.ModSetup_common_OID;
                int  timeMs;
                short  criticalSaveIntvalSecs;
                short[ ]  loop_length;
                short[ ]  loop_sep;
                byte  gDay;
                byte  gMonth;
                short  gYear;
                byte[ ]  laneAdj;
                byte  clss;
                byte  spdMph;
                byte  modemType;
                byte  roamingHour;
                byte  initialise;
                byte  difSensitivity;
                short  siteNo;
                byte[ ]  siteName;
                byte  initTime;
                byte  deviceType;
                short[ ]  bikeRatio;
                byte  dvdtWindow;
                byte  cycleAxles;
                byte  bMask;
                short  cycleTimeout;
                byte  captureLoop;
                byte  extVBV;
                byte  showEvent;
                byte  logOn;
                byte  showVBV;
                byte[ ]  releaseDate;
                byte[ ]  version;
                short  cfSize;
                byte  logdata;
                byte  bindata;
                short[ ]  bikeRatioMax;
                int  vehCount;
                short  pw_max;
                short  pw_min;
                byte  sigTail;
                byte  cycleAxlesMax;
                byte  vbvFileOpen;
                byte  dailyGSMHour;
                byte  faultCode;
                short  hourlyGSMmins;
                short[ ]  spare;
     }
     dynamic  dynamic_obj  =  new  dynamic();
     class  dynamic
     {
               //Fixed size = false
                static final int OID = SysOID.ModSetup_dynamic_OID;
```

```
                short  txSubsectorSize;
                byte[ ]  sendVBV;
                short  no_of_records_tx;
                int  start_sector;
                int  end_sector;
                byte  initialise;
                int  lastTxSector;
                int  lastBinSector;
                short  crc;
     }
}
```

5.3.6.4 TrafficData

```
class TrafficData
{
     td  td_obj  =  new  td();
     class  td
     {
               //Fixed size = false
                static final int OID = SysOID.TrafficData_td_OID;
                byte  gDay;
                byte  gMonth;
                byte  gYear;
                int  count;
                int  startTime;
                byte[ ]  vbvData;
                byte  instances;
     }
}
```

5.3.6.5 VbvBins

```
class VbvBins
{
     vbv  vbv_obj  =  new  vbv();
     class  vbv
     {
               //Fixed size = false
                static final int OID = SysOID.VbvBins_vbv_OID;
                short  archive;
                int  count;
                short  speed;
                short  pulse_width;
                short  gap;
                short  headway;
                short  chasisHeight;
                byte  lane;
                byte  axles;
                byte  class_id;
                byte  dir;
                byte  gDay;
                byte  gMonth;
                short  gYear;
                int  timeMs;
                byte  classMatch;
```

```
                byte  suspect;
                int  length;
     }
}
```

5.3.7 SP4/AXD40 Specific

5.3.7.1 ModuleState

```
class ModuleState
{
     sys  sys_obj  =  new  sys();
     class  sys
     {
               //Fixed size = false
                static final int OID = AppOID.ModuleState_sys_OID;
                int  validCode;
                short  criticalSaveIntvalSecs;
                byte  extVBV;
                byte  clss;
                byte[ ]  siteName;
                byte  rolling;
                byte  modemType;
                byte  roamingHour;
                byte[ ]  laneAdj;
                byte[ ]  onThreshold;
                byte  scanCount;
                byte  difSensitivity;
                byte  bMask;
                byte  dvdtWindow;
                short  siteNo;
                byte  tailSig;
                short[ ]  bikeRatio;
                byte  crcCheckmins;
                short  hourlyGSMmins;
                byte  dailyGSMHour;
                short[ ]  loop_length;
                short[ ]  loop_sep;
                byte  spdMph;
                byte[ ]  straddle;
                byte  straddleMs;
                byte  spareCRCed;
                short  crcFixedSetup;
                int  currTicks;
                byte  gDay;
                byte  gMonth;
                short  gYear;
                byte  logOn;
                byte  fileOpen;
                byte  read_only;
                byte  vbvFileOpen;
                byte  logdata;
                byte  bindata;
                byte  activePort;
                byte  showVBV;
                byte  showEvent;
```

```
                int  vbvCount;
                int  lastVBVCountSend;
                int  lastTxSector;
                byte  lastTxsubsector;
                short  txSubsectorSize;
                int  lastBinSector;
                short  binsectorOffset;
                short  fileBufferIn;
                short  no_of_records_tx;
                byte  captureLoop;
                byte  blockLivetoScreen;
                int  writeCycles;
                int  preState_Sector;
                short  resetCount;
                byte  ledOnSecs;
                byte  faultCode;
                byte  modem;
                short  crcForcedReset;
                byte  sendVBV;
                byte  gsmAutoPowerCount;
    }
    file  file_obj  =  new  file();
    class  file
    {
               //Fixed size = false
                static final int OID = AppOID.ModuleState_file_OID;
                byte[ ]  File_Name;
                short  start_cluster;
                int  start_sector;
                int  Fat_OpenCluster;
                byte  file_number;
                int  Fat_OpenSec;
                int  end_sector;
                int  file_size;
                int  file_size_start;
                byte  open;
    }
    fat  fat_obj  =  new  fat();
    class  fat
    {
               //Fixed size = false
                static final int OID = AppOID.ModuleState_fat_OID;
                byte  Num_Fats;
                int[ ]  Fat_Start;
                int  P_Start;
                byte  Secs_Cluster;
                short  Bytes_Sector;
                short  Secs_Fat;
                int  Root_Start;
                short  Res_Secs;
                short  crc;
    }
}
```

5.3.8 SP4/SLC4 Specific object composition

5.3.8.1 SlpState

```
class SlpState
{
     sys  sys_obj  =  new  sys();
     class  sys
     {
               //Fixed size = false
                static final int OID = AppOID.SlpState_sys_OID;
                int  validCode;
                short  criticalSaveIntvalSecs;
                byte  extVBV;
                byte  clss;
                byte[ ]  siteName;
                byte  rolling;
                byte  modemType;
                byte  roamingHour;
                byte  scanCount;
                byte  difSensitivity;
                byte  bMask;
                byte  dvdtWindow;
                short  siteNo;
                short[ ]  loop_length;
                short[ ]  bikeRatio;
                byte  dailyGSMHour;
                short  hourlyGSMmins;
                byte  spdMph;
                byte  straddleMs;
                byte[ ]  laneAdj;
                byte  spareCRCed;
                short  crcFixedSetup;
                int  currTicks;
                byte  gDay;
                byte  gMonth;
                short  gYear;
                byte  logOn;
                byte  fileOpen;
                byte  read_only;
                byte  vbvFileOpen;
                byte  logdata;
                byte  bindata;
                byte  showVBV;
                byte  showEvent;
                byte  captureLoop;
                byte  blockLivetoScreen;
                short[ ]  avgSpeed;
                short[ ]  avgPW;
                short[ ]  avgLength;
                byte  activePort;
                int  vbvCount;
                int  lastVBVCountSend;
                int  lastTxSector;
                byte  lastTxsubsector;
                short  txSubsectorSize;
                int  lastBinSector;
```

```
                short  binsectorOffset;
                short  fileBufferIn;
                short  no_of_records_tx;
                byte  tailSig;
                byte  ledOnSecs;
                byte  faultCode;
                int  writeCycles;
                int  preState_Sector;
                short  resetCount;
                short  crcForcedReset;
                byte  modem;
                byte  gsmAutoPowerCount;
                byte  sendVBV;
                byte  spare1;
                byte  spare2;
    }
    file  file_obj  =  new  file();
    class  file
    {
               //Fixed size = false
                static final int OID = AppOID.SlpState_file_OID;
                byte[ ]  File_Name;
                short  start_cluster;
                int  start_sector;
                int  Fat_OpenCluster;
                byte  file_number;
                int  Fat_OpenSec;
                int  end_sector;
                int  file_size;
                int  file_size_start;
                byte  open;
    }
    fat  fat_obj  =  new  fat();
    class  fat
    {
               //Fixed size = false
                static final int OID = AppOID.SlpState_fat_OID;
                byte  Num_Fats;
                int[ ]  Fat_Start;
                int  P_Start;
                byte  Secs_Cluster;
                short  Bytes_Sector;
                short  Secs_Fat;
                int  Root_Start;
                short  Res_Secs;
                short  crc;
    }
}
```

Chapter 6 MULTIDROP NETWORK

E ach unit runs multidrop MASTER/SLAVE protocol. These are packet based conforming to the packet described in section 5 above.

The setup is simple, put the appropriate link for the lane mask and just connect them together. Each individual slaves should be setup appropriately with MicroDial or a terminal.
Multidrop two wire (distance up to 4KM length) to each unit (A-A, B-B) . The MASTER device will hold all traffic data, individual slave device will hold their own as backups.
MASTER should also host GSM modem.

6.1 Address links & Lane mapping

The MASTER or SLAVE mode operation is selected by the address links. Hence each unit enable the appropriate protocol. Table below shows the link configuration to address mapping.

Within the network each unit must be set with a unique address.

- If no links are connected, device assumes "STANDALONE" mode. This shuts down the RS485 receiver and transmitter to save power.(Where there is a requirement for 16 lanes for SP4and 32 lanes for SP4/SLC4, a separate firmware is required which will need to uses this STANDALONE address)
- If all links are connected, it assumes the MASTER role. MASTER device always assumes lane 1 & 2
- All other link positions assumes for slave devices. This leaves 6 slave addresses with the links. (1..6 ids with 3 bit address).

The table below shows the link position and the lane mapping.

Description	*Jumper configuration lk1 lk2 lk3	SP4/SLC4 Address	SP4Address	SP4/SLC4 Lane mapping	SP4/AXD4 Lane mapping
Stand alone	0 0 0		23	1 to 4	1 to 2
MASTER	1 1 1		16	1 to 4	1 to 2
1st Slave	0 1 1		17	5 to 8	3 to 4
2nd Slave	1 0 1		18	9 to 12	5 to 6
3rd Slave	0 0 1		19	13 to 16	7 to 8
4th Slave	1 1 0		20	17 to 20	9 to 10
5th Slave	0 1 0		21	21 to 24	11 to 12
6th Slave	1 0 0		22	25 to 28	13 to 14

*Jumper configuration as viewed from the front. 0 – no link fitted, 1 – link fitted

6.2 Operational Settings

There are no operational settings. The master and each slave device must be setup separately as required.

Once connected to MicroDial send the update time command to the Master device. Time and date will be automatically broadcast by the master devices synchronising all slave devices.

The master will automatically link to each of the slave devices and start polling the instance it is connected to the slave.

6.3 Time & Date Broadcast synchronisation

The master device will automatically synchronise the time and date of all the slave devices. A time & date broadcast is issued when the time or date is changed in the Master device, or every midnight.

6.4 Data storage

The master will save the VBV data of all the slave units sequentially. In addition each slave device will save its own data. Each unit can buffer on the network queue up to 10 vehicles.

6.5 Polling rate vs. Vehicle rate

Each slave will be polled at least once in every 0.6 seconds. If all the units are connected the polling rate is improved. This is because a poll-reply is normally 30ms whilst a poll – no reply cycle is 100ms.

Each slave unit is able to store up to 10 vehicles in the network queue. This means the system will be able to cope at the worst case vehicle rate of (10/(0.6) 16 vehicles per second on both lanes or 8/vehicle/second per lane. It is practically impossible to get 8 vehicles per second on a lane.

Chapter 7 THE MICRODIAL SOFTWARE

7.1 Overview

The MicroDial Software provides support for Local and Remote installation, database, telemetry, data retrieval, site reports as well as the creation and analysis of reports.

Chapter 8 INSTALLATION

8.1 Classifier Installation

Qty	Part Number	Description
1		Outstation unit.
1		Compact Flash card
1		USB cable
1		GSM Modem
1		Modem Serial cable
1		PSU

8.2 Procedure

1. Remove packing and ensure that unit shows no signs of physical damage..
2. Record the serial number and location of the unit.
3. Plug the Loop Interface to Terminal Blocks cable into the rear of the unit. Secure each individual screw connection.
4. Ensure loop activity led reflect vehicle over the loop.

Chapter 9 COMMISSIONING AND TESTING

When the unit is first installed with a formatted compact flash card, the unit will enter into default initialisation mode. The 4 led's will flash every 1 second, this indicates that the unit is operating in default initialisation mode.

When date is set the unit will exit the default mode and enter into operational mode. All the setting will now assume default values.

The user should set the time and check the essential parameters before leaving the site, these are listed below.

9.1 Essential Parameter check

The following list the essential parameter check. The essential check may be done using a dump terminal (such as hyper terminal) or using MicroDial.

The following section deals with using a dump terminal. Refer to the MicroDial manual for auto commissioning and installing procedures.

- Connect the USB cable to the unit.
- Set the baud rate to 57600, 1,1 with no handshake

9.1.1 Date

Check date is correct.

9.1.2 Time

Check time is correct

9.1.3 Loop Separation

(For SP4Only)

Check loop separation is correct.

Loop separation is the gap between the leading and trailing loop

9.1.4 Loop Length

Check loop length is correct. Loop length is the longitudinal length of the loop, in line with the flow of the traffic.

9.1.5 Bike Ratios

For SP4

BIKE_RATIO

For SP4/SLC4

BIKE_RATIO12

BIKE_RATIO34

Bike Ratio's should not be less the 100. The default value is 120.

If bike ratios are not set correctly small cars and high chasis 4WD vehicles may be classed incorrectly.

9.1.6 Classification

Ensure that correct classification is selected.

9.1.7 Site name & Site number

Ensure that the site name and site number is setup

9.1.8 Speed measurement scale

Ensure that speed measurement scale is apprpriate, ie Mph or Kph.

9.1.9 Log file

Check the log file is opened or binary vehicle record file is open.

9.2 Site Operational check

Using the terminal type

VBV 3

Press the small front panel switch enable led indicators.

Wait for vehicles to pass to ensure that the led's are correctly indicating the presence of the vehicle as it passes over the loop.

Check the terminal shows the vehicle by vehicle line. Ensure that the vehicle length, speed and class are correct.

For advance site operational check, MicroDial is required which will display the signature profile of the vehicle.

Chapter 10 MAINTENANCE

10.1 Storage

The unit can be stored without any time limitation. There are no volatile components such as internal battery. When in storage it is recommended that the connectors are covered to avoid dust build up. This can be normally done by leaving the unit with a compact flash card connected and with all the loop and PSU connectors (as supplied) connected. USB and serial should be appropriately covered.

The internal PCB is conformal coated to provide enviromental robustness, to operate in wet and humid conditions.

10.2 Operational

Each unit is sophisticated electronic equipment using the latest chipset. The PCB's are clipped into the case, using no wire looms. It is a single board device incorporating the very latest family of integrated components.

Normal procedure should be to return failed unit or replace the PCB.

Chapter 11 TROUBLESHOOTING

11.1 Scope

11.2 Communications and Connections

11.3 Loops

Chapter 12 TECHNICAL SPECIFICATION

Physical	
Dimension	Width 170mm x Length 150mm x Height 40mm
Weight	Approx 500g
Case	4mm thick high impact polymer
Impact Protection	Drop from 5m
Ingress Protection	IP54
Operational	
MTBF	100000 Hours
Input Voltage	4v-18v
Current	18mA (typical) Variable on CPU frequencies 10mA, 15mA, 18mA or 32mA depending on the CPU frequency. The current is derived using Kingston CF cards.
Scanning Rate	Typical 1ms for 12mHz Variable to 10mS (depends on CPU frequency)
CPU Frequency	Std 12 MHz. 1.5Mhz, 6Mhz, 24Mhz
Real Time Clock	32khz crystal clock,
Temperature	-20C to +70 Further extended temperature range available on request.
Storage & Retrieval	
CF Card memory	256Mb, 512MB, 1GB, 2GB
File System	FAT16
Data Retrieval	Transfer Formats: ASCII Transfer Reliable Packet based transfer (CRC-16 bit) Methods CF card reader using PC/PDA GSM Dial-up connection GPRS data streaming POTS-Modem dial-up connection Direct Serial to PDA Direct USB to PC
Transfer Rates	GSM 9600 – 6600 line by line traffic data per minute. Direct or 57.6 POTs modem the figure is typically 12000/minute. File Transfer rate i 9600 - 57.6K

Storage	ASCII Text file Compact Binary Vehicle Record file. All files are readable using PC/PDA
Communications Interfaces	
CF	ATA- Interface
Serial	RS232 – Full handshake, DTE
USB	PC Connect, Driver known to Windows
RS485	Network interface, Up to 8 unit interconnectivity over 4km range using CAT5 twisted cable.
Network Address	
3-bit Internal Link	Each unit has firmware base address. The actual unit address is Base address + 3 bit internal link address. Allowing 8 unique addresses for each class of device.
Firmware	
Application	Flash upgradeable. All generic applications are Freely issued. Customised are locked to a particular hardware on client request.
Hardware Image	Internal upgrade, on return to factory
SP4Accuracy	2 –Lane Dual loop classifier
Count	>99%
Length	+/-3% with 95% Confidence
Speed	3% (Typically +/- 1MPH)
Axles	+/- 1 Axle 75% Confidence (Loop turns of 4 increases accuracy)
SP4/SLC4 Accuracy	Single Loop classifier
Count	>99%
Length	+/-8% with 95%Confidence Speed Range 10-75MPH
Speed	5%(Typically +/-2%) within 10-75 MPH
Axles	+/-1 Axle 60% Confidence.
C400 Accuracy	Cycle Counter
Segregated Count	>99%